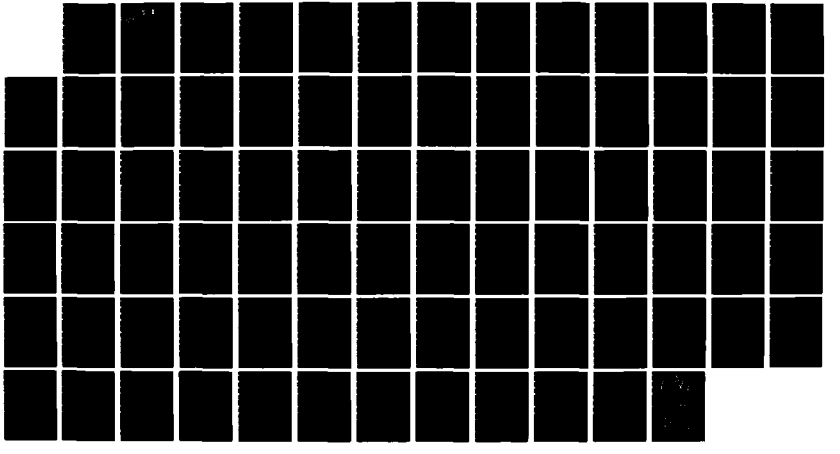
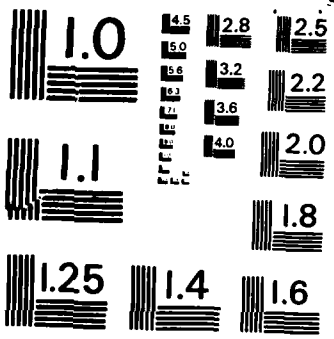
AD-A166 895 SECURE COMMUNICATIONS PROCESSOR STOP RELEASE 21(U) 1/1
DEPARTMENT OF DEFENSE FORT GEORGE G MEADE MD COMPUTER
SECURITY CENTER S J PADILLA ET AL. 23 SEP 85
UNCLASSIFIED CSC-EPL-85/001 F/G 17/2 NL





CSC - EPL - 85/001

(2)

FINAL EVALUATION REPORT OF SCOMP

Secure Communications Processor

STOP Release 2.1

23 September 1985

SECURITY CLASSIFICATION OF THIS PAGE

AD-A166 895

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release: Distribution Unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) CSC-EPL-85/001	5. MONITORING ORGANIZATION REPORT NUMBER(S) S227,782
6a. NAME OF PERFORMING ORGANIZATION Department of Defense Computer Security Center 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State and ZIP Code) 9800 Savage Road Ft. George G. Meade, MD 20755-6000	7b. ADDRESS (City, State and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT NO.
11. TITLE (Include Security Classification) Final Evaluation Report, Secure Communications Processor (SCOMP) STOP Release 2.1	
12. PERSONAL AUTHOR(S) Padilla, Steven J.: Benzel, Terry * * MITRE Corporation	
13a. TYPE OF REPORT Final 13b. TIME COVERED FROM _____ TO _____	14. DATE OF REPORT (Yr., Mo., Day) 85/09/23 15. PAGE COUNT
16. SUPPLEMENTARY NOTATION	
17. COSATI CODES FIELD / GROUP / SUB. GR.	18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) Trusted Computer System Evaluation Criteria SCOMP STOP Release 2.1 Verification EPL Honeywell DoDSCS A1

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The Department of Defense Computer Security Center (DoDCSC) was established in January 1981 to encourage the widespread availability of trusted computer systems for use by facilities processing classified or other sensitive data.

In the fourth quarter of FY82, Honeywell Information Systems, Inc., requested that the DoDCSC evaluate the SCOMP system, a commercially available operating system. SCOMP is a combination of hardware and software that provides the state of the art in Computer Security.

The security features provided by STOP Release 2.1 of the SCOMP System were evaluated against the requirements specified by the Department of Defense Trusted Computer System Evaluation Criteria dated 15 August 1983.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☒ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code) 22c. OFFICE SYMBOL

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

SECURITY CLASSIFICATION OF THIS PAGE

Team Members

Department of Defense Computer Security Center
9800 Savage Rd., Ft. Meade, MD

P. Farmer
P. Olson
S. Padilla
K. Wilson

MITRE Corporation
P.O. Box 208, Bedford MA

T. Benzel
J. Francis
D. Juitt
L. Scott
D. Tavilla

902nd MI Group, INSCOM
Ft. Meade, MD

D. Tischhauser

TRW Corporation
Los Angeles, CA

G. Short

Acknowledgement is given to the following individuals for their contributions to this evaluation: M. Brooks, D. Drake, H. Egdorf, J. Glass, V. Gligor, B. Hartman, T. Kilheeny, J. Kopp, J. Makey, R. McKercher, L. Smith, T. Taylor, G. Wagner, D. Wills, and P. Woodie.

CSC-EPL-85/001
Library No. S227,782

FOREWORD

This publication, Secure Communications Processor (SCOMP) STOP Release 2.1 Final Evaluation Report, is being issued by the Department of Defense Computer Security Center under the authority and in accordance with DoD Directive 5215.1, "Computer Security Evaluation Center." The purpose of this report is to document the results of the formal evaluation of Honeywell's SCOMP System. The requirements stated in this report are taken from the Department of Defense Trusted Computer System Evaluation Criteria dated 15 August 1983.

Approved:

ELIOT SOHMER
Chief, Standards and Products

23 September 1985

Table of Contents

1. EXECUTIVE SUMMARY

The Department of Defense Computer Security Center (DoDCSC) has evaluated the security mechanisms of the Honeywell Secure Communications Processor (SCOMP) SCOMP Trusted Operating Program (STOP) Release 2.1. This evaluation was completed using the requirements of the Department of Defense Trusted Computer System Evaluation Criteria (the Criteria) dated 15 August 1983.

The DoDCSC has determined that the SCOMP STOP Release 2.1 satisfies all the requirements of the Criteria Class A1 and therefore SCOMP STOP Release 2.1 has been assigned a Class A1 rating.

The Class A1 rating implies that the system provides mandatory (labeled) protection as well as discretionary (need-to-know) protection. The system does extensive auditing, and provides an unforgeable (trusted) communication path between the users and the system. The distinguishing characteristic of this class is the assurance derived from formal specification and verification of the security mechanisms.

This class of systems has been subjected to extensive testing and all implementation flaws discovered during testing have been corrected. However, no guarantee is made that the system is free of all implementation flaws.

2. INTRODUCTION

2.1 BACKGROUND

The DoDCSC was established in January 1981 to encourage the widespread availability of trusted computer systems for use by facilities processing classified or other sensitive information. In order to assess the degree of trust one could place in a given computer system, the Trusted Computer System Evaluation Criteria were written. The Criteria establish specific requirements that a computer system must meet in order to achieve a predefined level of trustworthiness. To determine the level in the Criteria at which a system should be placed, the system must be evaluated against the Criteria by a DoDCSC sponsored evaluation team. The Criteria are arranged hierarchically into four major divisions of protection, each with certain security relevant characteristics. These divisions are in turn subdivided into classes.

After examining the system in question, the evaluation team submits a report describing its findings. Three types of reports are prepared: a developmental evaluation, a preliminary assessment, and a formal evaluation. A developmental evaluation is produced based on a vendor's design for either security enhancements to an existing system or for a new trusted product. Included in a developmental evaluation is an in-depth examination of design plans. Since the evaluation is based on design documentation and information supplied by the vendor, the evaluation involves no "hands on" experience with the system. The preliminary assessment is usually done as a precursor to a formal evaluation and would emphasize aspects of the system that need to be improved in order to achieve a higher rating. As with the developmental evaluation a preliminary assessment would not require any "hands on" exposure to the system. The DoDCSC restricts distribution of both the developmental evaluation and the preliminary assessment reports.

The formal evaluation is assumed to be the evaluation of a system that will not undergo any additional changes. A formal evaluation requires "hands on" testing (e.g., functional and possibly penetration testing). Also, once a vendor has agreed to submit a system for a formal evaluation, the vendor may not withdraw it from the evaluation. The final report will be a summary of the evaluation and will include the final evaluation class assignment, and a description of the strengths and weaknesses of the system. This summary evaluation will be made public.

In August 1982, the Department of Defense Computer Security Center (DoDCSC) began, at the request of Honeywell Information

Systems (HIS), a formal product evaluation of the Honeywell product called the Secure Communications Processor (SCOMP). The objective of this product evaluation was to rate the SCOMP system against the DoDCSC Trusted Computer System Evaluation Criteria (the Criteria) and to place it on the Evaluated Products List (EPL) with the final rating. This report presents the results of that evaluation.

The system which was evaluated is a modified Honeywell Level 6 minicomputer (Model 43) enhanced by a hardware Security Protection Module (SPM), running STOP Release 2.1 software which consists of a security kernel package and trusted software, and associated documentation. A complete list of the software, hardware, and documentation are included in Appendices A and B.

2.2 SYSTEM CONFIGURATION

The test sites for the SCOMP system evaluation were the Honeywell Information Systems in McLean, VA, and The MITRE Corporation in Bedford, MA. Equipment included in the test configuration for each site is listed in Appendix B along with the software modules included in the Trusted Computing Base.

It should be noted that one security aspect of the SCOMP system is configuration dependent. The trusted recovery process requires two disk drives to repair the kernel files after a system crash. Absence of the second disk means that the trusted recovery process must include either reloading the disk from a backup disk (thereby losing the most recent work done on the system), or sending the pack to Honeywell or to another SCOMP system site that has a two-disk configuration. For further information, see Section 4.3.1.5, Trusted Recovery.

2.3 OVERVIEW

The remainder of this report is divided into sections as described herein. Section 3 contains a brief description of the SCOMP system architecture, with emphasis on the security-relevant features of the hardware and software mechanisms. It is intended as background information for the reader, but should not be considered a complete architectural description, which is given in the references.

Section 4 discusses in detail the comparison of the SCOMP system with the requirements listed in Section 4.1 of the Criteria. Section 4.1 was used because the SCOMP system was targeted as an A1 system.

Section 5 is a section containing comments of evaluators of the system. These comments do not affect the rating of the system but are given as useful observations of areas which are not necessarily covered by the Criteria.

3. DESCRIPTION OF THE SCOMP AND ITS SECURITY MECHANISMS

3.1 BACKGROUND

This section presents a brief architectural overview of the SCOMP system and its security features. It is included only to give the reader sufficient background for this report and is not intended to be complete. For a comprehensive discussion of the SCOMP hardware and software consult the Honeywell SCOMP system specifications as listed in Appendix A.

The SCOMP system consists of a commercial Honeywell Level 6 minicomputer enhanced by a Security Protection Module (SPM), special purpose security kernel software, trusted software, and an untrusted SCOMP Kernel Interface Package, SKIP. The security relevant architectural features that are used in the SCOMP system are based on concepts developed in Multics. They include the use of a hierarchical domain or ring structure, descriptor control of resources, and a segmented virtual memory. A primary access control mechanism in the SCOMP system is the use of descriptors. Descriptors are four-word data elements created in software by the kernel and used for mediation by the SPM.

3.2 SCOMP HARDWARE COMPONENTS

The hardware base for the SCOMP system is a Honeywell Level 6 (Model 43 or Model 53) minicomputer. The Level 6 is enhanced by adding several new central processor instructions, and a hardware SPM. In addition, the memory management unit (MMU) of the standard Level 6 has been replaced by the Virtual Memory Interface Unit (VMIU) of the SPM. These additions in conjunction with a software security kernel provide mediation and isolation while minimizing system degradation.

The Honeywell Level 6 is a bus-structured, 16-bit minicomputer consisting of a family of standard modules that plug into the bus. The bus provides a common communication path among all functional components, operates asynchronously, and supports all memory, command, and interrupt operations. There are three types of bus-connectable (plug-in) units in all Level 6 systems: central processors, input/output controllers, and memories. One or more of each type, up to a maximum of 23 units, can be included in the system configuration. All of the elements are attached to the bus, and all transfers (data, interrupts, and instructions) between them take place on the bus at a transfer rate of six to eight megabytes per second. The asynchronous bus operation allows components of varying speeds to operate efficiently within the same system [2].

The Level 6 is an execution domain machine. Execution domain machines provide a privileged mode and one or more unprivileged modes of processor execution. A mechanism which is implemented to distinguish between procedures operating in different execution domains is referred to as a protection ring. The term ring was coined by Bob Graham [3] to illustrate the notion of inclusion of privileges; i.e., inner rings include the privileges of the outer rings.

The SCOMP system provides four rings of execution. The rings can be considered as being arranged concentrically with ring 0 the innermost ring and ring 3 the outermost. Ring 0 is the most protected and therefore possesses the most privilege; ring 3 is the least protected. In the SCOMP system, rings 0 and 1 are treated as equally privileged and provide the kernel domain, ring 2 is the trusted software and operating system domain and ring 3 is the user domain. The SCOMP, like most execution domain machines, includes the concept of privileged instructions or operations; such operations can only be executed by a process operating in the required ring. Some examples of privileged instructions are those that modify virtual memory mapping registers and those that set the domain of execution for a process. Note that such instructions all belong to the set of instructions which must be executed by the kernel.

In an execution domain machine the address space of a process consists of a number of virtual memory objects each of which has been assigned to a set of execute and read rings and to a set of write rings. The basic virtual memory object in the SCOMP system is a segment which is 0-2K words in size. Every executing process has a ring number associated with it. This number is referred to as the current ring of execution. Before an executing process can access any object, the SPM compares the current ring number with a set of three ring bracket numbers. The ring bracket numbers are stored in the descriptor for the object and indicate the range of rings of execution from which the object may be accessed in write, read, and execute mode. The relationship between the process's ring number, the segment's ring brackets, the access modes recorded in the descriptor, and the requested access mode, determines whether the access will be allowed or denied. Access control through the use of rings and ring brackets is discussed in more detail in [1].

3.3 THE SECURITY PROTECTION MODULE

The major security mechanism that has been added to the Honeywell Level 6 is the hardware SPM. The primary functions of the SPM are the performance of access checks and the mapping of virtual references to physical references. Functionally, the SPM

is interspersed among the system elements and provides mediation for Central Processing Unit (CPU) references to memory, CPU references to I/O devices, and I/O device references to memory. The active element (CPU or I/O device) requests access to objects by a virtual address. The SPM hardware then uses the virtual address, together with a special process identifier (called a Descriptor Base Root or DBR), to find the appropriate object descriptor. This descriptor is then used by the SPM to translate the virtual address and to verify that the active element has the authority to perform the requested action.

The SPM utilizes descriptors for access control and address translation. Access to the physical resources of the system can only be obtained through descriptors. Descriptors contain the logical access permissions (e.g., read, write, execute) and the necessary data to map a virtual reference to a physical reference. The logical access permissions are those that are allowed based on the security attributes of the subject and the object.

In the SCOMP system each process has its own virtual address space, which can be described as the set of objects (e.g., segments) accessible by the process at a given time. An object is made accessible to a process when a descriptor for the object is added to the table of active object descriptors for the process. All effective program addresses formed by a process are virtual addresses and must be translated into physical main memory addresses.

Another security feature of the SCOMP system is that the concept of virtualization has been extended to include I/O devices. Descriptors are also used to define I/O devices. As with memory objects, a device is made accessible to a process when a descriptor for it is added to the process' active object table. The process then accesses the device using a virtual device address which is translated by the SPM into an effective access to a physical I/O device. In addition, I/O devices, channels, and Direct Memory Access (DMA) transfers to and from physical memory are similarly checked via the SPM. Thus, all I/O in the SCOMP system is performed through the use of descriptors. The effect of this use of virtual I/O in SCOMP system is a uniform subject-object access model that provides complete mediation of all accesses to memory and I/O, thereby allowing users to control their own I/O in a secure manner.

3.4 RING STRUCTURE

3.4.1 Ring Brackets

Ring brackets denote the range of allowed rings in which different modes of access are valid. This range is based on the access rules described by the segment's descriptor. The values in the segment's descriptor fields R1, R2, and R3 represents the highest (least privileged) ring of execution from which a process may write, read, or execute the segment. A segment's write bracket is defined as ring 0 through the ring value in R1; its read bracket is defined as ring 0 through the ring value in R2; its execute bracket is defined as the ring values in R1 through R2; and its call bracket is defined as the ring values in R1 through R3. For example, a kernel segment that is callable by a user process would have ring bracket values of 0,0,3 for R1, R2, R3. These values, in conjunction with the proper access privileges, would indicate that only the kernel could write and read the segment and any process executing in ring 0 through ring 3 could call it. On the other hand, a user segment would have the values 3, 3, 3 for R1, R2, and R3, which would indicate that any process, with the proper privileges, executing in ring 0 through ring 3 could read or write the segment but only ring 3 processes could call or execute the segment.

3.4.2 Cross Ring Movement

Processes can change their current ring of execution through the use of three new instructions for cross ring movement which have been added to the Level 6 instruction set [4]. The instructions are the Call, the Return, and the Argument Addressing Mode Instruction, and are discussed below.

3.4.2.1 Call and Return

The SPM recognizes Call and Return requests from an executing process. The Call order is similar to a transfer request except that the SPM has the ability to change the current ring number to a lower value. The Return is similar to a standard transfer, except it allows for an increase in current ring number. Calls are used to transfer to an inner ring procedure in order to accomplish a more privileged operation than that allowed in the current ring of execution. Returns are used to return from an inner ring procedure back to an outer ring. The only requirements for the Return instruction are that the ring returned to is specified and that the specified ring is of lower or equal privilege. Return to an inner ring is always prohibited.

It is not sufficient to restrict which segments may be called (via ring brackets); there must also be a means of specifying a location in the segment which is a valid entry point. This has been implemented by allowing only location zero of the segment to be a valid entry point. Thus, the SPM will verify that the offset of the virtual address is zero before changing the ring of execution for a Call instruction.

The Return instruction is used to transfer from an inner ring of execution to an outer ring. The Return instruction is used for this type of transfer rather than the Call because if calls were allowed to be made from an inner ring, possible security violations could occur. Access checking for the Return instruction is simpler than for the Call instruction.

3.4.2.2 Argument Addressing Mode

Cross ring movement can pose a particular security problem in regard to argument validation. This is because the process in the outer ring will supply arguments or argument pointers as part of its call to the inner ring. However, the inner ring has more privilege, and thus it may access data not accessible to the outer calling ring. Therefore, to avoid any misuse of this greater privilege, the inner ring must validate the caller's right to the arguments supplied in the call. To solve this problem, a new addressing-mode instruction, called Argument Addressing Mode (AAM), has been added. The AAM instruction allows a called procedure to access passed arguments at the privilege level of the caller.

3.5 INPUT/OUTPUT

All I/O in the SCOMP system is virtualized. This approach utilizes descriptors for mediation of access, similar to the use of memory descriptors for memory access mediation. This approach to controlling I/O in a secure manner has several advantages. Because all mediation is performed in hardware (i.e. the SPM), I/O instructions need not be executed in the kernel. This allows the device drivers to be outside of the kernel which makes the kernel smaller, simpler, and easier to verify. In addition, placing the device drivers outside of the kernel provides more flexibility for adding and changing the system device configuration. Finally, this approach offers distinct performance advantages over systems where I/O is performed via kernel calls.

In the SCOMP system devices are treated as special purpose processes which operate asynchronously with the initiating process. When a device is created (by the system administrator),

the device access information is attached to a configured logical device. In order to close potential information channels only one process may map a device at a time, and only one logical device may be active per physical channel.

3.5.1 Device to Memory

There are two means of performing device to memory I/O, premapped I/O and mapped I/O. The basic difference between these two depends on the type of information which is contained in the Direct Memory Access (DMA) device. This is because a DMA device, once initiated, controls the series of data transfers to/from memory. The SPM handles both types of I/O, and at initiation of the I/O uses the information within the I/O descriptors to determine which flow is applicable. In premapped I/O, the SPM mediates both the device and the access to the memory resource involved in the transfer and then initiates the transfer by setting up the device controller with absolute addresses. That is, the SPM interprets and translates the memory addresses at the initiation of I/O, and then the device subsequently uses absolute addresses without the intervention of the SPM. In mapped I/O, the device controller is initiated with a virtual memory address and each request by the device to memory is intercepted and mediated by the SPM. Mapped I/O requires a unique identifier code which is imbedded in the starting address of the device controller by the SPM. The purpose of this identifier is to enable the SPM to detect the source of the request and locate the proper descriptor. All devices operate in one of the two modes. The MT bit of the device's I/O descriptor contains the information for determining which mode the device operates in. In general random access fast I/O devices (e.g. disks) operate in premapped mode, while slower devices such as terminals operate in mapped mode.

3.5.2 I/O Address Translation

The SPM mediates all processor to I/O references. When the processor makes a reference to an I/O device, the address sent over the bus is intercepted by the SPM and is handled as virtual address. The SPM translates the virtual address into a physical device address through a series of look-ups in the descriptor tables. The device is then set up with the physical address and the transfer is made.

3.6 THE KERNEL

The kernel is the primary software security mechanism of the SCOMP system. In addition to implementing the reference monitor, the kernel performs the basic operating system functions,

resource management, process scheduling, memory management, trap and interrupt management, and auditing.

The kernel manages and controls access to the basic resources of the system. The resources fall into three categories or types of kernel objects: processes, memory segments, and devices. Every kernel object possesses a unique identifier, access control information, and status data. The unique identifier associated with each object is 64 bits in size, and the identifier remains constant during the life of the object. Access control information consists of the security level and category set, the integrity level and category set, and discretionary information in the form of an access control list that includes up to 9 entries. Three of those entries are set to the owner of the object, the owner's group, and a group for all others (world). The attributes associated with each entry are read, write, execute, and null permissions. The security and integrity levels are eight hierarchical classifications and the category sets consist of 32 separate compartments.

In addition to the three types of objects (processes, segments, and devices) supported at the kernel interface level, the kernel supports subtypes of each. Each user has a subtype list which can contain 5 entries per object type. The use of subtypes was included to provide kernel support for the construction of secure systems which implement additional discretionary access control policies, beyond the owner, group, other mechanism.

The security kernel provides 38 callable functions for accessing and modifying the kernel objects. These functions are known as kernel gates and allow for the creation and deletion of objects, mapping (inclusion in the process's address space) and unmapping of segments or devices, wiring (retention in memory) and unwiring of segments, getting or setting status, interprocess communication (IPC), reading and setting of the system clock, and user I/O.

The hierarchical file system is implemented outside the kernel to simplify the kernel verification and to use the hardware efficiently. The highest level of abstraction at which the kernel manipulates objects is that of a segment. Files are created, maintained, and managed by the SCOMP Kernel Interface Package(SKIP). Files are composed of segments and have a single security level. All non-kernel I/O is unprivileged; i.e., I/O instructions can be issued outside of ring 0, without requiring a kernel call for every I/O command issued. A process initiates I/O by mapping the desired I/O device (via the map gate of the kernel), and then performs its own I/O to the device. The

kernel's only involvement is to pass on to the process all interrupts occurring on the device. The physical I/O commands are executed in the user's ring of execution.

3.7 TRUSTED SOFTWARE

Users, system administrators, and operators access certain kernel functions through the trusted software. Trusted software consists of these functions which are security-related, yet do not execute within the kernel. Trusted software is unique in that it uses special security kernel privileges to perform trusted functions. This software is considered trusted for one of two reasons: either it has the ability to violate one of the security or integrity properties enforced by the kernel, or it performs functions on which the system's ability to enforce its security policy depends. For example, one function in trusted software is the database editor that creates the user access database. If the software creates this database incorrectly, the kernel cannot enforce its policy on such user actions as login.

Trusted software is divided into four functional categories: User Services that provide the user interface; Trusted Operation Services that provide the system operator functions; Trusted Maintenance Services that provide the system administrator functions; and two Trusted SKIP Services, "Delete Upgraded" and "Set Segment Ownership."

3.8 SCOMP KERNEL INTERFACE PACKAGE

The SCOMP system does not provide a complete operating system emulator on top of the security kernel. The original design for the SCOMP system interface employed a Bell Labs UNIX (tm) emulator. The SCOMP system has been implemented with an interface to the secure environment which provides the users with a higher level interface to the primitive functions of the kernel. In order to achieve a given operating system environment on the SCOMP system, the user can write a command process, or shell, and develop the set of interface subroutines which would map the desired environment to the proper combination of SKIP calls.

SKIP is composed of two types of modules: gates and subroutines. The major portion of SKIP resides in a group of protected ring 2 gate segments which are wired into memory. This portion of SKIP is accessed through SKIP gate calls which execute in ring 2. The remaining portion of SKIP consists of a library of subroutines which are linked with the user programs and execute in ring 3. The SKIP gates and subroutines provide file and process management, and non-kernel device I/O facilities. Due

to the fact that SKIP does not function as part of the Trusted Computing Base (TCB), it is not included in the evaluation.

3.9 SUMMARY

The SCOMP System was designed with security as the primary goal. In particular, special hardware was developed to operate in conjunction with specialized software to provide a secure computing environment. Isolation of users is achieved through the use of virtual memory, and by descriptor controlled access to the system resources which is enforced by the hardware ring mechanism. An important security feature of the SCOMP system is the use of virtual I/O, which was designed to allow users to efficiently and securely control their own I/O. Finally, the SCOMP Kernel Interface Package allows users to use the SCOMP system in varied application environments.

4. DETAILED CORRESPONDENCE BETWEEN CRITERIA AND SCOMP

4.1 SECURITY POLICY

Division A systems are characterized by the use of formal security verification methods to assure that the mandatory and discretionary security controls employed in the system can effectively protect classified and other sensitive information stored on or processed by the system. Extensive documentation is required to demonstrate that the TCB meets the security requirements in all aspects of design, development, and implementation.

This section of the report deals with the requirements listed on pages 42 through 50 of the Criteria [5]. Each quoted requirement will be followed by an evaluation of that aspect of the system and the conclusion arrived at by the evaluation team.

4.1.1 Discretionary Access Control

Requirement

> The TCB shall define and control access between named users and named objects (e.g., files and programs) in the ADP system. The enforcement mechanism (e.g., access control lists) shall allow users to specify and control sharing of those objects. The discretionary access control mechanism shall, either by explicit user action or by default, provide that objects are protected from unauthorized access. These access controls shall be capable of specifying, for each named object, a list of named individuals and a list of groups of named individuals with their respective modes of access to that object. Furthermore, for each such named object, it shall be possible to specify a list of named individuals and a list of groups of named individuals for which no access to the object is to be given. Access permission to an object by users not already possessing access permission shall only be assigned by authorized users.

Evaluation

All SCOMP kernel objects, i.e., processes, segments, and devices, have discretionary access control attributes associated with them. The information stored for each object consists of an access control list that includes up to 9 entries, three of those entries are set to the owner of the object, the owner's group, and a group for all others (world). The attributes associated

with each entry are read, write, execute, and null permissions. To create an object on the SCOMP system a minimum set of access attributes must be specified, i.e. owner permission, group permission, world permission. This leaves 6 entries free for additional identifiers which may be added at a later time.

Files on the SCOMP system are made up of segments and have associated with them the access control attributes of the segments. The owner of a file is capable of changing the access to the file through the use of the file access modifier (fam) [6] command. Fam permits the addition and deletion of user numbers or group numbers to/from the access control list for the file. When a group number or user number is added to an access control list the owner must specify one or more of read, write, execute access, or null access. Any user number or group number that has null specified as its access attribute is restricted from any direct access to the contents of the file. The fam command may be used to review the current access to any file. The fam command displays a list of group numbers, user numbers and their associated access permissions. In order to determine current access to the file it is necessary to review the access control list as well as the list of user numbers associated with each group in the access control list.

Users are normally assigned to their default group upon completion of the login sequence. A user with set_user_group privilege is capable of changing groups by issuing a trusted set group (sg) [6] command. The set group command will ascertain that the user is a member of the specified group before permitting the user to change to that group. (The group access authentication database is maintained by the security administrator using the group access authentication database editor (ga_edit) [6] command.)

The ACL consists of entries for the owning user, the owning group, other specific users, other specific groups, and the world. Access for any user to some object is determined by the first ACL entry found which applies to that user. The ACL is searched in the following order: owning user, other specific users, owning group, other specific groups, world.

Conclusion

The SCOMP system meets the A1 requirements for discretionary access, both for defaults and for named users and named groups. The implementation of access control lists on all objects provides the named users and groups that is required. The defaults are provided by the create object gates which require that a minimum set (owner, group, world) of access controls

associated with the object be specified by the user when the object is created.

4.1.2 Object Reuse

Requirement

> When a storage object is initially assigned, allocated, or reallocated to a subject from the TCB's pool of unused storage objects, the TCB shall assure that the object contains no data for which the subject is not authorized.

Evaluation

Secondary storage objects are segments which consist of 1 to 8 256-word (512-byte) blocks on disk. Whenever a segment is deleted, its disk blocks are zeroed and placed on the free list. When a new segment is created or when an existing segment is expanded in size, the new blocks are allocated from the free list.

Unlike secondary storage, primary memory is not zeroed upon deallocation, rather, it is zeroed before allocation. When information is moved from secondary storage to primary memory, the information in primary memory is overwritten.

Conclusion

The SCOMP system purges data from secondary storage prior to its reuse as an active storage object. While primary memory is not similarly purged upon deallocation, purging does occur prior to allocation of new segments. Furthermore, the SCOMP hardware does ensure that each page of primary storage is overwritten with the correct page from secondary storage which in turn was previously allocated from the free list of zeroed storage blocks.

The SCOMP system meets all requirements for object reuse.

4.1.3 Labels

Requirement

> Sensitivity labels associated with each ADP system resource (e.g., subject, storage object) that is directly or indirectly accessible by subjects external to the TCB shall be maintained by the TCB. These labels shall be used as the basis for mandatory access control decisions. In order to import non-labeled

data, the TCB shall request and receive from an authorized user the security level of the data, and all such actions shall be auditable by the TCB.

Evaluation

Labels are an integral part of the SCOMP system. The labels on the SCOMP system are composed of two parts, a security part, and an integrity part [7]. The first part is made up of a single byte to represent one of eight possible security levels, and four bytes that represent any combination of up to 32 categories. The integrity label is constructed the same as the security label. This evaluation concerned itself primarily with the security label, however, the integrity label plays an important role in assuring that the TCB is tamperproof and therefore it is mentioned at this time. All mandatory access control decisions are based upon comparison of these labels.

All files are made up of segments and all segments have an associated label. All devices have a label associated with them except for mass storage devices (disks) which have labels on the segments stored on the device. The labeling of SCOMP objects is described in greater detail in the following sections.

The TCB attaches the label associated with the channel to the data received over that channel. All channels except for disk are single level channels. Data that is imported via a (single-level) channel has attached the label that is associated with the channel at the time that it is received by the TCB. Users running at operator or higher integrity may change the level associated with non-terminal devices through the use of the set_device_access (sda) command [6]. An audit record is generated whenever the level of a device is changed.

Conclusion

The SCOMP system meets the requirement for labels.

The following requirements are related to this area and describe in greater detail the workings of labels on the SCOMP system.

4.1.3.1 Label Integrity

Requirement

Sensitivity labels shall accurately represent security levels of the specific subjects or objects

with which they are associated. When exported by the TCB, sensitivity labels shall accurately and unambiguously represent the internal labels and shall be associated with the information being exported.

Evaluation

Sensitivity labels on the SCOMP system are maintained by the Kernel and the Trusted Software. All users are assigned their default working level when they initially login to the system. The process that is created for them by the TCB is labeled according to the working level. Default working levels are managed by the security administrator via the user access authentication database editor (ua_edit) [6]. (Sensitivity labels associated with users is further discussed in the Subject Sensitivity Labels Section of this report.) All objects on the system also have a sensitivity label associated with them. The labels on the processes and the labels on the objects of the system are of the same form and structure [7]. There is a one-to-one mapping of bits between the subject label and the subject level.

Since disks are the only multi-level devices, they are the only devices which store trusted labels. When labels are exported to disk the structure and format of each label is maintained.

Conclusion

The SCOMP system meets the requirement for label integrity by providing a large enough label to directly map to all security levels and categories, and by applying the same label structure to all subjects and objects.

4.1.3.2 Exportation of Labeled Information

Requirement

The TCB shall designate each communication channel and I/O device as either single-level or multilevel. Any change in this designation shall be done manually and shall be auditable by the TCB. The TCB shall maintain and be able to audit any change in the current security level associated with a single-level communication channel or I/O device.

Evaluation

The SCOMP system designates all communication channels and I/O devices as single level devices with the one exception of mass storage devices (disks). Disks may be designated as either single level or multilevel by the operator each time the disk is physically mounted. The disk drive that holds the SCOMP object code defaults to multilevel and cannot be changed by the operator. The status of a disk drive (multilevel or single-level) may only be set once after bringing the device on-line. To change the status of the drive the device must be physically taken off-line and then brought back on-line. When the operator designates a disk drive as multilevel or single-level an audit record is generated.

On all single-level devices, except terminals, the operator must issue the set_device_access (sda) command to change the current level of the device [6]. Devices may change levels as often as the operator sees fit but each time that this occurs an audit record is generated.

Terminals are a special type of device. Their level is tied to the working level of the user. The level of the terminal is changed each time the working level of the user is changed, and the new level is displayed on the user's screen. Whenever the working level of a user is changed an audit record is generated. (See Subject Sensitivity Labels below).

Conclusion

The SCOMP system meets the requirement for designating all channels as single or multilevel and auditing any change to this status.

4.1.3.3 Exportation to Multilevel Devices

Requirement

> When the TCB exports an object to a multilevel I/O device, the sensitivity label associated with that object shall also be exported and shall reside on the same physical medium as the exported information and shall be in the same form (i.e., machine-readable or human-readable form). When the TCB exports or imports an object over a multilevel communication channel, the protocol used on that channel shall provide for the unambiguous pairing between the sensitivity labels and the associated information that is sent or received.

Evaluation

The only devices in the SCOMP system which may be designated as multilevel are disks. Whenever a disk is designated as multilevel the label associated with data is sent to the disk and stored on the disk with the data. The label on the disk has the same structure [7] as the internal label used by the TCB to make mandatory access decisions. Whenever data is imported from a multilevel disk the sensitivity label is sent with the data. This label is then used for any further access decisions until the time that the data is removed from the control of the TCB (i.e., the data is exported or deleted from the TCB).

Conclusion

The SCOMP system meets the requirement for Exportation to Multilevel Devices by storing, sending, and receiving the sensitivity labels with the data.

4.1.3.4 Exportation to Single-Level Devices

Requirement

> Single-level I/O devices and single-level communication channels are not required to maintain the sensitivity labels of the information they process. However, the TCB shall include a mechanism by which the TCB and an authorized user reliably communicate to designate the single security level of information imported or exported via single-level communication channels or I/O devices.

Evaluation

The single-level devices on the SCOMP system include terminals, printers, tape drives, communication ports, and disk drives when specified as single-level. All of these devices have associated with them a label which specifies the current level of the device.

Whenever data is received from the device the data is labeled with the label associated with the device. Once the data has been labeled it remains at that level until an authorized user requests that the label be changed. Any changes to the device label will not affect data that has already been received from that device.

Whenever a request is made to send data to a device (export data) the level of the data is checked against the current level of the device. If the level of the data is higher than the level of the device the data is not sent and an audit record is generated (see auditing).

To change the current level of a single-level device a privileged user (operator or administrator) executes the set_device_access (sda) command [6]. Execution of this command will cause an audit record to be generated. Any attempt by an unauthorized user to change a device level will result in a failed attempt which is audited. As was previously mentioned, this does not apply to terminals.

The working level of the terminal is changed by setting the working level of the current user of the terminal via the set_access_level (sl) command [7]. Any such change generates an audit record. (See Subject Sensitivity Labels below).

The level of the printer is also handled differently since its level is changed by the trusted printer daemon. (See device labels below).

Conclusion

The SCOMP system meets the requirements for single-level devices by assigning a label to each single-level device and using that label for the data that is imported and exported through the device.

4.1.3.5 Labeling Human-Readable Output

Requirement

The ADP system administrator shall be able to specify the printable label names associated with exported sensitivity labels. The TCB shall mark the beginning and end of all human-readable, paged, hardcopy output (e.g., line printer output) with human-readable sensitivity labels that properly* represent the sensitivity of the output. The TCB shall, by default, mark the top and bottom of each page of human-readable, paged, hardcopy output (e.g., line printer output) with human-readable sensitivity labels that properly* represent the overall sensitivity of the output or that properly* represent the sensitivity of the information on the page. The TCB shall, by default and in an appropriate manner, mark other forms of human-readable output (e.g., maps, graphics) with human-readable sensitivity labels that properly* represent the sensitivity of the output. Any override of these marking defaults shall be auditable by the TCB.

Evaluation

The SCOMP system provides the security_map database editor (sm_edit) command [6] that permits the administrator to specify and change the printed name for each level and category on the system. The name associated with each level and category includes up to 19 printed characters [6]. (Note that this reference incorrectly indicates a maximum of 20 characters.)

The SCOMP system marks the top and bottom of each page and the beginning and end of hardcopy printer output with the security level and security categories. The integrity level and integrity categories are not printed. The SCOMP system only supports single-level files and single-level output on each page.

---

* The hierarchical classification component in human-readable sensitivity labels shall be equal to the greatest hierarchical classification of any of the information in the output that the labels refer to; the non-hierarchical category component shall include all of the non-hierarchical categories of the information in the output the labels refer to, but no other non-hierarchical categories.

The label, therefore, represents the highest security level and security category of the data contained on each page. This default cannot be altered by the user. There is no capability for anyone to redefine the page size. Therefore, there is no overriding of the default labels.

Conclusion

The SCOMP system meets the requirement for labeling human readable output by providing two mechanisms. The first is an editor that can be used by the administrator to specify the printed names associated with each security level and category. The second mechanism is labeling the beginning and end of all printed files with the security label of the file. The SCOMP system also prints the same label at the top and bottom of each page. Since labeling of printed output cannot be overridden there is no need to audit the override of the defaults.

4.1.3.6 Subject Sensitivity Labels

Requirement

> The TCB shall immediately notify a terminal user of each change in the security level associated with that user during an interactive session. A terminal user shall be able to query the TCB as desired for a display of the subject's complete sensitivity label.

Evaluation

The SCOMP system does not change the security level of a user automatically. The user must initiate the action to change working levels. Therefore, the working level cannot change without the user's knowledge. The set_access_level (sl) command [7] allows the user to query the TCB for the current working level as well as to change that level. In addition, the user's current working level is displayed upon entering the trusted environment.

Conclusion

The SCOMP system meets the requirements for subject sensitivity labels by providing two functions. The first function provided is displaying the current working level upon initiation of the trusted environment. The second function is providing a trusted command that can be executed at any time by the user to determine the current working level.

4.1.3.7 Device Labels

Requirement

The TCB shall support the assignment of minimum and maximum security levels to all attached physical devices. These security levels shall be used by the TCB to enforce constraints imposed by the physical environments in which the devices are located.

Evaluation

All devices on the SCOMP system are considered by the TCB to be single level devices except for the disks. For single level devices, the maximum level equals the minimum level. The operator or administrator may reset the level of a single level device using the set_device_access (sda) command [6].

Terminals are a special case of single level device. They have associated with them a maximum level which is a site selectable option set by Honeywell prior to system delivery. The current level of the terminal is determined by the subject security level of the user. This level must be less than the predefined maximum level for that terminal. The minimum level associated with any terminal is system low.

Printers are also a special case of single-level device. They are controlled by a trusted printer daemon which varies the level of the printer according to the level of the output as long as the level of the data being printed is less than the maximum level associated with the printer. The minimum for the printers is also system low.

The maximum and minimum levels for the disks are determined by the maximum and minimum levels set by the security administrator for each filesystem on the disk when it was created, using the make_filesystem (mkfsys) command [6]. Thus the maximum level for a kernel disk is the highest of the maximum levels of the several filesystems that may be on the disk. The minimum level for the disk is the lowest minimum level associated with any filesystem on the disk. These levels cannot be changed dynamically. A new filesystem must be created by the system administrator in order for these levels to be changed.

Conclusion

The SCOMP system meets the requirement for device labels by providing maximum and minimum levels on all devices. The administrator controlled maximum on terminals and printers

provides the capability to control access to sensitive data in an unrestricted environment.

4.1.4 Mandatory Access Control

Requirement

> The TCB shall enforce a mandatory access control policy over all resources (i.e., subjects, storage objects, and I/O devices) that are directly or indirectly accessible by subjects external to the TCB. These subjects and objects shall be assigned sensitivity labels that are a combination of hierarchical classification levels and non-hierarchical categories, and the labels shall be used as the basis for mandatory access control decisions. The TCB shall be able to support two or more such security levels. The following requirements shall hold for all accesses between all subjects external to the TCB and all objects directly or indirectly accessible by these subjects: A subject can read an object only if the hierarchical classification in the subject's security level is greater than or equal to the hierarchical classification in the object's security level and the non-hierarchical categories in the subject's security level include all the non-hierarchical categories in the object's security level. A subject can write an object only if the hierarchical classification in the subject's security level is less than or equal to the hierarchical classification in the object's security level and all the non-hierarchical categories in the subject's security level are included in the non-hierarchical categories in the object's security level.

Evaluation

The kernel enforces its mandatory access control policy through the use of the hardware security protection module (SPM), as described in [1]. This policy is applied to the three types of kernel objects: devices, processes, and segments. Sensitivity labels are assigned to each device, process, and segment.

All SCOMP system kernel objects have a security classification that is a combination of security level and categories, and integrity level and categories assigned to them. There may be as many as eight hierarchical security levels and 32 non-hierarchical security categories, and as many as eight

hierarchical integrity levels and 32 non-hierarchical integrity categories. Labels are described more completely in section 4.1.3.

Administrator integrity level, the highest integrity level available on the evaluated SCOMP system, is required to modify the database containing the trusted label names. It should be noted that this "security map" database is maintained at a security level such that all users of the system have read access to the database. Thus, all users of the system can know just what types of information might be stored in the system.

The security levels are 8 hierarchical classifications and the category sets are 32 separate compartments for both security and integrity. The security levels are set by the system security administrator and are contained in a protected system database. For processing of DOD classified data the levels can be set to: UNCLASSIFIED, CONFIDENTIAL, SECRET, and TOP SECRET. They are ordered such that, UNCLASSIFIED < CONFIDENTIAL < SECRET < TOP SECRET. Although 8 hierarchical integrity levels may be defined, typically only three are defined such that USER < OPERATOR < ADMINISTRATOR. The relationship between the security and integrity levels of a subject and an object determines whether the object is accessible to a subject. In particular, a subject can read an object only if the security level of the subject is greater than or equal to security level of the object and the security categories of the subject include those of the object and if the integrity level of the subject is less than or equal to the integrity level of the object and the integrity categories of the subject are included in those of the object. And a subject can write an object only if the security level of the subject is less than or equal to security level of the object and the security categories of the subject are included in those of the object and if the integrity level of the subject is greater than or equal to the integrity level of the object and the integrity categories of the subject include those of the object.

Conclusion

The SCOMP system meets the requirements for mandatory access control by assigning sensitivity labels to all subjects and objects, protecting the labels that are assigned, and using those labels to enforce an acceptable mandatory access control policy.

4.2 ACCOUNTABILITY

4.2.1 Identification and Authentication

Requirement

> The TCB shall require users to identify themselves to it before beginning to perform any other actions that the TCB is expected to mediate. Furthermore, the TCB shall maintain authentication data that includes information for verifying the identity of individual users (e.g., passwords) as well as information for determining the clearance and authorizations of individual users. This data shall be used by the TCB to authenticate the user's identity and to determine the security level and authorizations of subjects that may be created to act on behalf of the individual user. The TCB shall protect authentication data so that it cannot be accessed by any unauthorized user. The TCB shall be able to enforce individual accountability by providing the capability to uniquely identify each individual ADP system user. The TCB shall also provide the capability of associating this identity with all auditable actions taken by that individual.

Evaluation

The SCOMP system requires that each user enter a valid user name and password combination before anything else can be done. The user name and password may contain up to 15 characters each. Note that there is no minimum length on passwords. A limit is set on the number of tries that a user is allowed before a security violation occurs. When a violation occurs, the terminal associated with the violation is automatically locked out for a set amount of time and a message is sent to the system console. The default limit for invalid logon is 3 attempts and the default limit for terminal lockout is 1 minute. These two values are stored in the terminal configuration database and may be changed by the system administrator using the terminal_configuration database editor (tc_edit) command [6] to fit the needs of a particular site.

The passwords on the SCOMP system are encrypted and the encrypted password is stored in a user access database. (The strength of the encryption scheme was not evaluated as part of this evaluation.) The user access authentication database, which may contain as many as 255 unique entries, is maintained at the

highest security and integrity levels available on the system. This database is maintained by the system administrator using the user_access_authentication database editor (ua_edit) command [6].

Security administrators may grant users the privilege to change their passwords. Users change their passwords using the change_user_password (cup) command [7]. The administrator may change any user's password by using the change_user_password (cup) command [6].

An audit record is generated whenever any attempt is made to login to the system. After valid login all actions taken by a user are associated with that user's unique identification when the action is recorded in the audit file.

Conclusion

The SCOMP system meets all the requirements for identification and authentication. Each user is required to identify himself at logon and authenticate the logon using a password. The SCOMP system has a sufficient number of user-IDs to allow each user to be individually identified. The identification and authentication information is stored at system high so that only those users with proper privileges may access the information. The audit information associates user-IDs with all user actions mediated by the TCB.

4.2.2 Trusted Path

Requirement

> The TCB shall support a trusted communication path between itself and users for use when a positive TCB-to-user connection is required (e.g., login, change subject security level). Communications via this trusted path shall be activated exclusively by a user or the TCB and shall be logically isolated and unmistakably distinguishable from other paths.

Evaluation

The SCOMP system utilizes a secure attention key, which is implemented as the break key, to initiate a trusted path of communication between the user and the TCB. It is this combination of hardware and supporting software upon which the SCOMP system relies for its trust in the trusted path. The trusted path cannot be initiated via a program or without the user's knowledge because it uses a signal that must be generated by external hardware. Because of the ability of another computer

or intelligent terminal to simulate the break key function, sites must restrict the types of devices connected as terminals.

The trusted path is utilized for logon processing, logoff, process control, changing passwords, changing discretionary groups, downgrading of files, and changing the security level of the working user level and system devices. It is also used for operator and administrator commands. All user actions requiring the protection of a distinct user-to-TCB communication utilize the trusted path (break key). The TCB does not initiate any communication to the user via the trusted path. The system authentication to the user is provided by the knowledge that only the TCB can respond to the interrupt generated by the secure attention key. Since the SCOMP system does not support programming of a simulated break key it cannot be spoofed by using SCOMP software.

Conclusion

The combination of hardware and software of the SCOMP trusted path (break key) are impossible to simulate without additional hardware. All security critical functions of the user utilize the trusted path. The system meets the class A1 requirement for trusted paths.

4.2.3 Audit

Requirement

> The TCB shall be able to create, maintain, and protect from modification or unauthorized access or destruction an audit trail of accesses to the objects it protects. The audit data shall be protected by the TCB so that read access to it is limited to those who are authorized for audit data. The TCB shall be able to record the following types of events: use of identification and authentication mechanisms, introduction of objects into a user's address space (e.g., file open, program initiation), deletion of objects, and actions taken by computer operators and system administrators and/or system security officers. The TCB shall also be able to audit any override of human-readable output markings. For each recorded event, the audit record shall identify: date and time of the event, user, type of event, and success or failure of the event. For identification/authentication events the origin of request (e.g., terminal ID) shall be included in the audit record. For events that introduce an object into

a user's address space and for object deletion events the audit record shall include the name of the object and the object's security level. The ADP system administrator shall be able to selectively audit the actions of any one or more users based on individual identity and/or object security level. The TCB shall be able to audit the identified events that may be used in the exploitation of covert storage channels. The TCB shall contain a mechanism that is able to monitor the occurrence or accumulation of security auditable events that may indicate an imminent violation of security policy. This mechanism shall be able to immediately notify the security administrator when thresholds are exceeded.

Evaluation

Extensive audit information is maintained for both the kernel and trusted software in separate files. Audit records are generated for the following events:

segment_creation	segment_map
segment_deletion	segment_unmap
segment_access_change	segment_access_violation
segment_owner_change	
device_creation	device_map
device_deletion	device_unmap
device_access_change	device_access_violation
device_owner_change	
process_owner_change	process_access_violation
process_privilege_change	process_space_runout
process_subtypes_change	
mount_space_runout	loader_error
branch_block_runout	trusted_delete_error
data_block_runout	fam_error
disk_error	
change_default_level	set_group
cancel_terminal_lockout	set_level
change_password	shutdown
login	logout
audit_admin_cmd	audit_operator_cmd

The SCOMP system also has a display audit file (daf) tool that provides selectable criteria for examining the audit data.

This tool is protected from tampering by assigning a special integrity (audit) category to the tool. This category keeps all normal users from being able to write to or modify the tool. The selection criteria provided by this tool includes such things as security level, user name, and file identifier, among others.

The type of audit information to be recorded is designated by the administrator through the use of the system parameter database editor (param_edit) command [6].

The SCOMP system successfully audits eight of the twelve covert kernel channels identified by Honeywell [8]. The remaining four channels have very low estimated bandwidths. Timing delays have been added to the kernel software to decrease the bandwidth of the covert channels.

Each audit record contains the event type, the time the event occurred, the process that caused the event, the privileges of the process, and the mandatory and discretionary access information about the process that caused the event. Section 13 of the SCOMP Trusted Facility Manual [6], contains documentation of each audit record generated about the trusted software and the kernel.

The operator is automatically notified of the following: terminal lockout, disk errors, and unexpected interrupts from tape drive, printer, or diskette [6].

The audit information files are protected by the standard mandatory and discretionary access mechanism provided by the kernel. The files are maintained at operator integrity with a special integrity category (audit) and system high security. Users at system high security who have discretionary access permission may view and selectively search the audit file using the standard SCOMP kernel gate calls. The discretionary access permissions on the files may be set so that only authorized users may view the audit data.

Conclusion

The SCOMP system meets all of the requirements for auditing by being able to audit all security related events and being able to selectively view the data that has been collected.

4.3 ASSURANCE

4.3.1 Operational Assurance

4.3.1.1 System Architecture

Requirement

The TCB shall maintain a domain for its own execution that protects it from external interference or tampering. The TCB shall maintain process isolation through the provisions of distinct address spaces under its control. The TCB shall be internally structured into well-defined largely independent modules. It shall make effective use of available hardware to separate those elements that are protection-critical from those that are not. The TCB modules shall be designed such that the principle of least privilege is enforced. Features in hardware, such as segmentation, shall be used to support logically distinct storage objects with separate attributes. The user interface to the TCB shall be completely defined and all elements of the TCB identified. The TCB shall be designed and structured to use a complete, conceptually simple protection mechanism with precisely defined semantics. The mechanism shall play a central role in enforcing the internal structuring of the TCB and the system. The TCB shall incorporate significant use of layering, abstraction and data hiding. Significant system engineering shall be directed toward minimizing the complexity of the TCB and excluding from the TCB modules that are not protection-critical.

Evaluation

The SCOMP system architecture satisfies the requirement for a separate protected domain through the use of a protection ring mechanism that is implemented in hardware and firmware. The SCOMP TCB resides in the most privileged rings, which protect the TCB from interference or tampering. The ring mechanism and the hardware Security Protection Module (SPM) mediation of per process virtual memory provide process isolation.

The TCB is implemented using a top-down design. It is made up of independent modules, each of which has a single entry point. The kernel software is implemented in the Pascal programming language, and the trusted software is implemented in the C programming language. The hierarchical nature of the kernel and trusted software is described in the program

functional flow trees included in the Part II specifications [9,10].

The principle of least privilege is strictly adhered to throughout the design and implementation of the TCB. The proper use of privileges is enforced by the kernel and protection ring mechanism. Each function in the SCOMP TCB possesses and uses the minimum set of privileges necessary for its functionality.

The kernel supports three types of objects: processes, segments, and devices. Each of these objects is distinguished by a unique 64-bit identifier that never changes for the life of the object in the system. The kernel maintains access information and status data on each object in the system. The access information is used to control mandatory as well as discretionary access to the objects. The status information varies depending on the object type.

The SCOMP system TCB is described in [11]; which also identifies the elements of the user interface. This user interface is formally specified in the FTLS for the kernel and trusted software (see the discussion on Design Specification and Verification 4.3.2.2.) The TCB consists of:

a. the SCOMP Security Kernel, which is composed of hardware, firmware, and software elements;

b. the SCOMP Trusted Software, which is composed of individual processes that provide specialized services to users and processes; and,

c. users logged into the system at or above the operato. integrity level. (Note: All Operator/Administrator functions are security relevant and are considered to be internal to the TCB.)

The protection mechanism used in the SCOMP system is a reference monitor. The SCOMP system's implementation consists of a kernelized operating system, hardware enforced protection rings, and a special purpose SPM which mediates all resource requests. This reference monitor implementation is the fundamental security mechanism of the SCOMP system.

The SCOMP system protection ring and kernel gate mechanisms provide layering, data hiding and abstraction such that external functions requesting services of the TCB are provided only minimal information. The size and the complexity of the TCB has been minimized in order to allow formal verification techniques

to be applied. Review of the TCB design and implementation has demonstrated that only protection-critical functions are included in the TCB.

Conclusion

The SCOMP system architecture was designed to provide a basis for a secure system. The evidence presented by Honeywell demonstrates that the design of the SCOMP system meets all security requirements for system architecture.

4.3.1.2 System Integrity

Requirement

> Hardware and/or software features shall be provided that can be used to periodically validate the correct operation of the on-site hardware and firmware elements of the TCB.

Evaluation

Included with the SCOMP system is a Test and Verify package called SCMPAC. This package can be used to check the functioning of the hardware base of the SCOMP system. The SCMPAC performs extensive tests of main memory, disk drives, tape drives, and the CPU. The tests are flexible enough to allow the site to run only those tests that it feels are necessary as well as specify how thorough the tests should be. The system must be taken down before performing these tests, and then is brought up using the SCMPAC standalone disk pack.

The SCOMP system has some testing procedures built into it. These procedures are known as Quality Logic Tests. They are run every time a reboot is executed. These tests check all boards for correct functioning.

Conclusion

The system integrity requirement is met by the SCOMP system via the SCMPAC.

4.3.1.3 Covert Channel Analysis

Requirement

> The system developer shall conduct a thorough search for covert channels and make a determination (either by actual measurement or by engineering

estimation) of the maximum bandwidth of each identified channel. Formal methods shall be used in the analysis.

Evaluation

Honeywell performed formal covert channel analysis on the kernel and the trusted software. The covert channel analysis of the kernel was performed using the Multilevel Security (MLS) Tool of SRI's Hierarchical Development Methodology. This formal analysis discovered twelve covert channels. Bandwidth analysis of the known covert channels has shown that all channel bandwidths are within the guidelines presented in the Criteria. Furthermore, delays have been incorporated into the implementation in order to further minimize the bandwidth of all covert storage channels. Eight of the twelve covert channels are audited. The remaining four channels either have bandwidths that are acceptably low or can be closed by placing a suitable warning in the Trusted Facility Manual [6]. Covert timing channel analysis has shown that some well known timing channels [12] do exist in the SCOMP system, however it has been shown that these channels are extremely noisy and would be exceedingly difficult to exploit.

Information flow and covert channel analyses were performed on the trusted software. The method used for the analysis was based on the secure information flow techniques described in [13]. This formal analysis of the trusted software formal top level specification (FTLS) discovered one covert channel. The worst case bandwidth of this channel is well within the guidelines presented in the Criteria, and is less than the rate at which auditing is recommended.

Conclusion

Honeywell has conducted a thorough search for covert channels using formal methods. A determination of the maximum bandwidth of each identified channel has been made and all bandwidths are within the guidelines presented in the Criteria. The Honeywell analysis meets all Criteria requirements for covert channel analysis.

4.3.1.4 Trusted Facility Management

Requirement

The TCB shall support separate operator and administrator functions. The functions performed in the role of a security administrator shall be identified. The ADP system administrative personnel

shall only be able to perform security administrator functions after taking a distinct auditable action to assume the security administrator role on the ADP system. Non-security functions that can be performed in the security administration role shall be limited strictly to those essential to performing the security role effectively.

Evaluation

The SCOMP Trusted Facility Manual [6] contains separate sections that describe the operator and administrator functions. All operator functions are available to the administrator; however, none of the administrator functions are available to the operator. The separation of these users from unprivileged users is enforced by the integrity mechanism provided on the system. All operator and administrator functions require integrity higher than user. The integrity mechanism is used to constrain the operator and administrator from performing other than security related functions. Operators and administrators have no access to low integrity functions.

The operator and administrator can only perform their privileged functions after either login at the operator/administrator integrity level or after issuing the set_access_level (sl) command [6] and changing their level to operator or administrator. Both of these actions are audited by the TCB.

Conclusion

The Trusted Facility Management features of the SCOMP system and the documentation satisfy the Criteria at the A1 level.

4.3.1.5 Trusted Recovery

Requirement

Procedures and/or mechanisms shall be provided to assure that, after an ADP system failure or other discontinuity, recovery without a protection compromise is obtained.

Evaluation

The mechanism for a trusted recovery of SCOMP STOP Release 2.1 consists of the trusted software functions check and repair. The check function verifies the correct structure of each specified kernel filesystem. Filesystem errors can damage

security protection mechanisms, resulting in security violations. The repair function, also invoked by the trusted check [6] command, fixes any damage to the filesystem. It may be possible to restart the SCOMP system after a system failure without repairing the filesystem, but such a restart violates the system's security and is prohibited by the Trusted Facility Manual.

The trusted recovery procedure does require at least two disk drives. The check function runs on one disk drive and checks the filesystem on the other disk drive. If only one disk drive exists on the system, the check and repair functions cannot be performed.

Conclusion

The SCOMP system meets the A1 criteria for trusted recovery. Procedures are provided to allow suitable recovery after a system failure that causes damage to the kernel file system. The documentation that describes the procedures for such a recovery allow a system manager to determine the proper actions to be performed in most cases of failure.

4.3.2 Life Cycle Assurance

4.3.2.1 Security Testing

Requirement

> The security mechanisms of the ADP system shall be tested and found to work as claimed in the system documentation. A team of individuals who thoroughly understand the specific implementation of the TCB shall subject its design documentation, source code, and object code to thorough analysis and testing. Their objectives shall be: to uncover all design and implementation flaws that would permit a subject external to the TCB to read, change, or delete data normally denied under the mandatory or discretionary security policy enforced by the TCB; as well as to assure that no subject (without authorization to do so) is able to cause the TCB to enter a state such that it is unable to respond to communications initiated by other users. The TCB shall be found resistant to penetration. All discovered flaws shall be corrected and the TCB retested to demonstrate that they have been eliminated and that new flaws have not been introduced. Testing shall demonstrate that the TCB implementation is consistent with the formal top-level specification.

No design flaws and no more than a few correctable implementation flaws may be found during testing and there shall be reasonable confidence that few remain. Manual or other mapping of the FTLS to the source code may form a basis for penetration testing.

Evaluation

Security testing includes the areas of kernel functional testing, trusted software functional testing, and penetration testing.

Kernel Functional Testing - Honeywell has performed extensive functional testing on the SCOMP kernel. Documentation for the kernel tests appears in [14,15,16]. The kernel test requirements, test descriptions, and source code were reviewed. The evaluation team reviewed the test requirements to ascertain their completeness in testing all parts of the SCOMP kernel. The kernel test requirements were then compared with the test descriptions to ensure that each test fulfilled the test requirement. About one-third of the kernel test code was then checked against the test descriptions. There are over 500 separate test requirements, and each test program often satisfies more then one requirement. Sixty percent of the kernel tests can be automatically initiated by a driver program (called a regression run). The evaluation team re-ran all kernel tests that are in regression runs in order to validate the tests. In addition, the evaluation team modified three of the kernel tests to test different parameters and to examine the process Honeywell used for testing the kernel.

The tests were designed to execute all paths through the kernel code. The kernel tests run on a system containing only the kernel code without the trusted software or SKIP code, in a stand-alone environment. Each test is isolated from other processing on the system. No kernel test was run concurrently with any other test or any system activities. Exception conditions exercised by kernel tests include: outside process address space, invalid segment number, invalid partition, or bad subtype.

The detailed specifications for the SCOMP system were written according to MIL-STD 490. These Part I (B) specifications state the functional characteristics of SCOMP kernel functions. The kernel test requirements were designed from the kernel detailed specifications to execute every line of kernel code. The kernel implementation has been shown to be

consistent with the formal top-level specification. The results of the specification-to-code correspondence were used to generate flaw hypotheses.

Trusted Software Functional Testing - Honeywell has provided a test plan, test summary and test results for the Trusted Software [17,18,19]. These tests, although not as extensive as the kernel tests, do test valid entry and boundary conditions in the trusted software functions. The trusted software tests were not rerun, but the functions were extensively exercised during penetration testing.

Penetration Testing - The penetration effort followed the flaw hypothesis methodology [20]. Flaw hypotheses were derived from the source code, the formal top level specifications, the detailed specifications, the results of the formal specification-to-code mapping, the functional tests, the covert channel results, the user's manual, the trusted facility manual, and documentation of the SCOMP hardware. During the penetration period, approximately 70 flaw hypotheses were generated and prioritized according to likelihood of existence and security value. These hypotheses were reviewed by the penetration team and a list of 5 minor discrepancies between documentation and implementation was delivered to Honeywell. These discrepancies were corrected and the corrections were reviewed by members of the original penetration team.

Conclusion

The security mechanisms of the SCOMP system have been found to work as claimed. Testing demonstrates consistency with the formal top-level specification and meets the requirements of the Criteria. The penetration testing discovered no design flaws and very few minor implementation flaws. All discovered flaws were corrected.

4.3.2.2 Design Specification and Verification

Requirement

> A formal model of the security policy supported by the TCB shall be maintained that is proven consistent with its axioms. A descriptive top-level specification of the TCB shall be maintained that completely and accurately describes the TCB in terms of exceptions, error messages, and effects. A formal top-level specification of the TCB shall be maintained that accurately describes the TCB in terms of exceptions, error messages, and effects. The DTLS and FTLS shall

include those components of the TCB that are implemented as hardware and/or firmware if their properties are visible at the TCB interface. The FTLS shall be shown to be an accurate description of the TCB interface. A convincing argument shall be given that the DTLS is consistent with the model and a combination of formal and informal techniques shall be used to show that the FTLS is consistent with the model. This verification evidence shall be consistent with that provided within the state-of-the-art of the particular Computer Security Center-endorsed formal specification and verification system used. Manual or other mapping of the FTLS to the TCB source code shall be performed to provide evidence of correct implementation.

Evaluation

The SCOMP system was formally specified and verified using state-of-the-art specification and verification techniques. Two Computer Security Center-endorsed specification and verification systems, the Gypsy Verification Environment and the Hierarchical Development Methodology (HDM) were used. The security policy which the SCOMP system enforces is the DoD policy on multilevel secure computing [21]. The formal model of this security policy enforced by the SCOMP TCB is the Computer Security Center accepted Bell and LaPadula Model [22].

The descriptive top-level specification (DTLS) of the SCOMP system is comprised of the Part I and II specifications for the kernel, trusted software, and the SPM [9,10,23,24,25,26]. The Part I specifications describe all TCB functions (software and hardware) in terms of their input, processing, and output. The Part II specifications provide more implementation detail including module and data interfaces, structure, and traceability tables. Inspection of these specifications as part of the specification-to-code correlation process and penetration effort has determined that the specifications completely and accurately describe the TCB.

A formal top-level specification (FTLS) for the kernel [27] was developed using SPECIAL, the formal specification language of HDM. A formal top-level specification of the trusted software [28] was developed using Gypsy, the formal specification language of the Gypsy Verification Environment. These abstract specifications formally describe the TCB in terms of exceptions, error messages, and effects. Inspection of these specifications as part of the specification-to-code correlation process and penetration effort has determined that the specifications accurately describe the TCB.

The DTLS includes a complete description of the hardware and/or firmware portions of the TCB, and the FTLS includes the hardware and/or firmware components whose properties are visible at the TCB interface (see Appendix D).

The DTLS and the FTLS have been shown to be consistent with the Bell and LaPadula model. This demonstration was done informally through the construction of an interpretation of the Bell and LaPadula model for the SCOMP system [29]. The interpretation consists of a mapping between the formal model entities and the SCOMP TCB entities, and a mapping between the formal model rules of operation and the SCOMP TCB functions.

Formally, the kernel FTLS has been shown to be consistent with the information flow model developed by SRI [32,33]. The trusted software FTLS was formally shown to be consistent with function-specific security requirements.

Manual mappings were constructed from the FTLS to the user visible portions of the SCOMP system TCB. The kernel FTLS written in SPECIAL was mapped on a line by line basis to the Pascal implementation code for the kernel [30]. These mappings demonstrate that the code was derived from the specifications. Honeywell provided adequate justification for each line of code that did not directly correlate to the FTLS [34]. The trusted software FTLS written in Gypsy was mapped on a line by line basis to the C implementation code for the trusted software [31]. Again all lines of code which did not directly correlate were justified by Honeywell [35].

Conclusion

A complete and accurate description of the TCB is provided by the descriptive top-level and formal top-level specifications. These specifications include descriptions of the hardware components of the TCB as well as the software components.

The design of the SCOMP system TCB has been formally specified and verified using a combination of formal and informal techniques. The verification evidence was obtained by using two Computer Security Center-endorsed formal specification and verification systems. The formal model of the security policy supported by the TCB is the Computer Security Center accepted, Bell and LaPadula Model.

The design specification and verification evidence provided by Honeywell demonstrates that the SCOMP system fully meets all Criteria requirements of operational assurance for Design Specification and Verification.

4.3.2.4 Configuration Management

Requirement

> During the entire life-cycle, i.e., during the design, development, and maintenance of the TCB, a configuration management system shall be in place for all security-relevant hardware, firmware, and software that maintains control of changes to the formal model, the descriptive and formal top-level specifications, other design data, implementation documentation, source code, the running version of the object code, and test fixtures and documentation. The configuration management system shall assure a consistent mapping among all documentation and code associated with the current version of the TCB. Tools shall be provided for generation of a new version of the TCB from source code. Also available shall be tools, maintained under strict configuration control, for comparing a newly generated version with the previous TCB version in order to ascertain that only the intended changes have been made in the code that will actually be used as the new version of the TCB. A combination of technical, physical, and procedural safeguards shall be used to protect from unauthorized modification or destruction the master copy or copies of all material used to generate the TCB.

Evaluation

The Configuration Management Plan [36] outlines the methods and safeguards used by Honeywell to maintain the consistency of software, hardware, and supporting documentation as changes are made to the system.

The software source and associated documentation of the SCOMP system was developed and is currently maintained on a Honeywell Level 6 computer running a Honeywell version of Unix. There is a separate set of Unix directories for each release of STOP, which are controlled using Unix discretionary access controls. A second backup copy is maintained offline.

The software change procedure utilizes a configuration control board and a problem report database, and accounting is by Internal Software Note, a sequentially assigned number. The Unix diff command is used to compare versions of STOP and provides a line by line listing of the differences.

The hardware configuration management uses standard Honeywell procedures and includes a hardware control board and change testing. These procedures ensure that changes to the security relevant hardware are traceable and do not invalidate security assumptions made by the software. To ensure that the hardware changes do not affect the security provided by the software all hardware changes are reviewed by the software development staff before they are implemented.

Since development of the SCOMP system began before the Criteria was written, the requirement that configuration management be applied throughout the entire life-cycle of the system has been waived. The configuration management procedures described above are presently being used by Honeywell.

Conclusion

The SCOMP system configuration management plan does provide the necessary tools and safeguards required to maintain the security of the system. No tools for generation of systems from software are provided to the user sites since no source is provided. Honeywell generates all versions of the SCOMP system and uses its trusted distribution mechanism to deliver versions to sites. The tools to perform this function are at Honeywell and are maintained by them.

The SCOMP system configuration management plan meets all the requirements for the A1 criteria.

4.3.2.5 Trusted Distribution

Requirement

> A trusted ADP system control and distribution facility shall be provided for maintaining the integrity of the mapping between the master data describing the current version of the TCB and the on-site master copy of the code for the current version. Procedures (e.g., site security acceptance testing) shall exist for assuring that the TCB software, firmware, and hardware updates distributed to a customer are exactly as specified by the master copies.

Evaluation

Honeywell maintains the mapping between the master copy of the source code and site versions through their configuration control mechanism. When a release is approved for public distribution a backup copy is made and kept off-line. The master

copy is kept on-line and write permission is removed so that no further changes can be made.

When a site purchases a SCOMP system, a description of the desired configuration must be sent to Honeywell. A set of configuration files are then set up and the desired release, with the site specific configuration files, is generated. A check-sum algorithm is then applied to the executable code. The executable code is sent to the site along with a checksum generation program. The checksum that was originally generated is then sent to the site. The site can run the checksum generation program and compare the result with the checksum delivered through the mail. The two checksums provide a means whereby the site can ascertain that the system that they received was the same system that Honeywell sent to them.

These procedures are documented in the Configuration Management Manual [36] issued by Honeywell and the Trusted Facility Manual [6] Section 11.

Conclusion

The trusted distribution plan submitted for the SCOMP system meets the criteria for an A1 system.

4.4 DOCUMENTATION

4.4.1 Security Features User's Guide

Requirement

> A single summary, chapter, or manual in user documentation shall describe the protection mechanisms provided by the TCB, guidelines on their use, and how they interact with one another.

Evaluation

The first two sections of the SCOMP User's Reference Manual [7], provide an overview of the security protection mechanisms available on the SCOMP system and maintained by the TCB. These include: the underlying ring mechanism, the authentication mechanism, the secure attention key, the object subtype mechanism, the discretionary access control mechanism, the non-discretionary (or mandatory) security categories, integrity categories, hierarchical security levels, and hierarchical integrity levels. The information presented in the introduction on protection mechanisms is very general and assumes an understanding of multilevel security policy. While the manual

assumes a level of security-related knowledge on the part of the reader, it does provide a comprehensive explanation of the basic protection mechanisms of the system. This explanation is not presented in tutorial format; however, enough data is included to determine guidelines on the use of the protection mechanism.

Conclusion

The requirement for documentation on security features in a user's guide is met by the SCOMP User's Reference Manual.

4.4.2 Trusted Facility Manual

Requirement

> A manual addressed to the ADP system administrator shall present cautions about functions and privileges that should be controlled when running a secure facility. The procedures for examining and maintaining the audit files as well as the detailed audit record structure for each type of audit event shall be given. The manual shall describe the operator and administrator functions related to security, to include changing the security characteristics of a user. It shall provide guidelines on the consistent and effective use of the protection features of the system, how they interact, how to securely generate a new TCB, and facility procedures, warnings, and privileges that need to be controlled in order to operate the facility in a secure manner. The TCB modules that contain the reference validation mechanism shall be identified. The procedures for secure generation of a new TCB from source after modification of any modules in the TCB shall be described. It shall include the procedures to ensure that the system is initially started in a secure manner. Procedures shall also be included to resume secure system operation after any lapse in system operation.

Evaluation

The Trusted Facility Manual [6] is divided into 14 sections. The first two sections are very similar to the first two sections of the user's manual. They present the philosophy of protection implemented in the SCOMP system in a high level approach.

Section 3 describes those commands which are reserved for the operator. Section 4 describes the administrator commands. In both of these sections the manual presents a synopsis of the

expected interaction between the system and the privileged user. The synopsis is followed by a brief description of the proposed use of the command and any related information that may be useful to the person using the command. This section is followed by a description of the security requirements placed on the user, such things as the minimum security and integrity levels needed in order to use this command are given. The last section of each command is a summary of possible error messages and their meanings.

Section 5 presents warnings to the operator/administrator about precautions that need to be taken in order to run the system in a secure manner.

Section 6 describes the method for securely starting the system and the procedure to follow after a crash.

Section 7 describes the real time warnings that are posted to the system console. Each message is accompanied by a description of the cause of the message.

Section 8 describes the system gates that can be used to implement site specific system applications. The description of each gate follows the same format as the description of the operator/administrator commands in sections 3 and 4.

Section 9 describes the procedure to be followed by a site which desires to write and install a trusted process. It should be noted that the trusted processes can be privileged and that the rating described in this report does not apply to a system that has any trusted processes install other than those listed in section 3 and 4 of the trusted facility manual.

Section 10 presents a list of the modules that make up the TCB of the SCOMP system.

Section 11 describes the method for installing a new SCOMP system. The mechanism for trusted distribution of updates and site acceptance testing is presented.

Section 12 presents the structures available to systems programmers for use in assessing information about the system.

Section 13 presents a detailed description of each audit record as well as a brief overview of auditing in the SCOMP system.

Section 14 presents a list of error codes that are returned by calls to kernel gates.

Conclusion

Sections 8, 9, 12, and 14 are useful information for systems programmers trying to generate secure applications on the system. The other ten sections provide the information called for in the trusted facility manual requirement of the Criteria. The SCOMP system meets all the requirements of this criteria.

4.4.3 Test Documentation

Requirement

> The system developer shall provide to the evaluators a document that describes the test plan and results of the security mechanisms' functional testing. It shall include results of testing the effectiveness of the methods used to reduce covert channel bandwidths. The results of the mapping between the formal top-level specification and the TCB source code shall be given.

Evaluation

Beginning November 1981, status information on each individual test and on changes to the kernel has been maintained. The status information maintained on each test includes: the module tested, the primary gate tested, the date last tested, and a kernel problem number. When a kernel problem was detected it was assigned a number with information on the problem maintained separately. The date when a kernel module was changed was also noted. All this information helped to assure that corrections to kernel code did not cause subsequent problems.

The SCOMP Kernel Test Report [16] describes the testing procedure, philosophy, and results. Honeywell also supplied the information maintained on the kernel tests, the kernel test specifications, and the kernel test code. These documents demonstrate that a thorough job has been done in testing the kernel.

(1) Trusted Software Testing and Documentation - The trusted software provides services to the users, operators, and administrator. These services are in the form of trusted commands and database editors. In addition to the commands and database editors, the supported trusted software is responsible for secure initialization, startup, server, spooler, loader, file system maintenance, secure SKIP functions, and passing trusted software gate calls to the kernel gate.

Honeywell developed a Trusted Software Test Plan for the SCOMP [17] STOP Release 1.2. The plan, procedures, and results were applied to all releases since STOP Release 1.2. The plan and procedures have been improved slightly with the additional trusted software functions added with new releases. The documentation presented for STOP Release 2.1, The Trusted Software Test Report [18], includes the test plan, the test summary and the test results.

(2) Covert Channel Documentation - Honeywell provided two documents which describe the covert channel analysis performed on the system [8,37]. These two documents describe all the storage and timing channels that Honeywell detected. The documents also describe the methods used to limit the bandwidth of these covert channels.

(3) FTLS-to-Code Mapping - Honeywell provided several documents that describe the FTLS-to-Code Mapping [30,31,34,35]. These documents show the correlation between the functions in the FTLS and the functions in the source code. The functions have been mapped down to line numbers in the code. This correlation was provided for both the kernel [30] and the trusted software [31]. Justifications for the unspecified portions of the kernel were provided in [34]. Justifications for the unspecified portions of the trusted software were provided in [35].

Conclusion

The test documentation requirement of the Criteria is satisfied in the areas of kernel test documentation, kernel covert channel analysis, and kernel FTLS-to-code mapping. The requirements in the Criteria for trusted software test documentation, trusted software covert channel analysis, and trusted software mapping are also met. The combination of testing of the kernel and trusted software meets the requirements for testing of the TCB in the Criteria.

4.4.4 Design Documentation

Requirement

> Documentation shall be available that provides a description of the manufacturer's philosophy of protection and an explanation of how this philosophy is translated into the TCB. The interfaces between the TCB modules shall be described. A formal description of the security policy model enforced by the TCB shall be available and proven that it is sufficient to enforce the security policy. The specific TCB protection

mechanisms shall be identified and an explanation given to show that they satisfy the model. The descriptive top-level specification (DTLS) shall be shown to be an accurate description of the TCB interface. Documentation shall describe how the TCB implements the reference monitor concept and give an explanation why it is tamperproof, cannot be bypassed, and is correctly implemented. The TCB implementation (i.e., in hardware, firmware, and software) shall be informally shown to be consistent with the formal top-level specification (FTLS). The elements of the FTLS shall be shown, using informal techniques, to correspond to the elements of the TCB. Documentation shall describe how the TCB is structured to facilitate testing and to enforce least privilege. This documentation shall also present the results of the covert channel analysis and the tradeoffs involved in restricting the channels. All auditable events that may be used in the exploitation of known covert storage channels shall be identified. The bandwidths of known covert storage channels, the use of which is not detectable by the auditing mechanisms, shall be provided. Hardware, firmware, and software mechanisms not dealt with in the FTLS but strictly internal to the TCB (e.g., mapping registers, direct memory access I/O) shall be clearly described.

Evaluation

The manufacturer's philosophy of protection is documented in [38] and its translation into the TCB given in [11]. The interfaces between the TCB modules are described in the several Part II specifications, viz. [9,10,26]. A formal description of the security policy model (Bell and LaPadula) that is enforced by the TCB is given in [22] for the general case and Multics in particular in [39]. The Bell and LaPadula Model has been accepted by the Computer Security Center to model security policy [21] and to be consistent with its axioms. No SCOMP-unique document is required. An interpretation of the model for the SCOMP system is given in [29].

The specific TCB protection mechanisms are 1) protection rings, 2) SPM mediation of per user virtual memory, 3) minimum privilege for each TCB function, 4) integrity levels for users, operators, administrators, and security administrators, 5) individual trusted software processes for separate functions, and 6) ring gates and checks on parameter passing. The Part II Specifications previously referenced provide the necessary

documentation for satisfaction of this requirement. The explanation given to show that the TCB protection mechanisms satisfy the model appears in [29].

Section 3 of [11] describes the SCOMP TCB reference monitor implementation. An analysis of the Reference Monitor appears in Appendix C and concludes that the informal proofs that the SCOMP system implements the reference monitor concept are adequate.

The TCB implementation was shown to be consistent with the FTLS by specification to source code mappings [30,31,34,35]. TCB testing is documented in [14,15,16,17,18,19,40]. The TCB structure provided added assurance of the validity of the testing and helped to demonstrate the implementation of least privilege. The results of the covert channel analysis including conservative bandwidth estimates are presented in [8,37]. Auditable events, identified in Section 13 of [6], and the scheme of randomly selected delays on exception returns appear to satisfactorily limit the utility of the identified covert channels.

Finally, the internal TCB mechanisms that are not security related and hence not dealt with in the FTLS are described in the commercial Honeywell Level 6 documentation [41,42], and the SCOMP system unique specifications [9,10,26].

Conclusion

The design documentation requirement is met by a large collection of vendor-provided materials that are referenced in the preceding discussion and appear in the list of references.

5. Evaluators' Comments

5.1 Environment

Since the Criteria apply to application-independent or off-the-shelf products, a product evaluation does not consider an ADP system's operating environment when assigning a rating. A vital factor in most DoD operating environments is the degree of trust to protect information. The degree of trust is greatly increased with an inherently secure system as measured by the Criteria. A rating for a particular site should be a preliminary consideration in the acquisition of an ADP system.

The SCOMP system has an inherently high level of security that environmental security procedures can build upon. Although originally designed as a front-end processor for a communications network (hence its name), its expansion to a multipurpose minicomputer has greatly expanded its range of potential applications and environments. Again, it is emphasized that an ADP system's potential or targeted operating environment plays no role in the evaluation of that system.

5.2 Features

The SCOMP system provides many useful features that are not explicitly required by the Criteria. One such feature is the integrity mechanism. Integrity levels and categories can be used to further separate users and protect data. The integrity mechanism is capable of establishing an environment that cannot be modified by the general user community. This static environment provides a base upon which applications can be constructed and relied upon to function without undesirable modification. It also may be very useful to those who desire to protect the integrity of data that is placed on the system. While the integrity mechanisms of the SCOMP system were examined by the evaluation team, they were not subjected to formal analysis and penetration testing.

The areas of robustness and performance also were not considered as part of this evaluation. The team, through frequent use of the system, observed that these areas may present a concern in regard to the SCOMP system. Improvements to these areas were made throughout the evaluation but the evaluation was hindered due to problems in these areas.

5.3 Conclusion

In conclusion, the team feels that this system does provide the state-of-the-art security that would be expected of a system

rated A1. There are also many other features and capabilities that are provided with this system that are beyond the scope of this evaluation. It should be emphasized that the scope of this report is limited to the security provided by the system and does not include the suitability of the system for any specific application.

APPENDIX A

SCOMP VERSIONS AND DOCUMENTATION EVALUATED

SCOMP software version STOP 2.1 was the released version against which the evaluation (comprising verification, functional testing and penetration testing) was completed.

Documentation provided by Honeywell and examined as part of the evaluation effort is listed below. Unless otherwise stated all documents were published by Honeywell Information Systems, Inc., McLean, VA.

User Documentation

1) "SCOMP User's Reference Manual, STOP Release 2.1," 1 November 1984.

2) "SCOMP Trusted Facility Manual, STOP Release 2.1," 1 November 1984.

3) "SCOMP System Release Bulletin, Release 2.0," 29 May 1984.

4) "SCOMP System Release Bulletin, Release 2.1," 1 November 1984.

5) "SCOMP System Release Bulletin, Release 2.1, Addendum A," 2 November 1984.

Software Specifications

1) "Top Level Specification for SCOMP Kernel Software, Release 2.0," 29 June 1984.

2) "SCOMP Trusted Computing Base," 25 July 1984.

3) "SCOMP Interpretation of the Bell-LaPadula Model," October 25 1984.

4) "Detail Specification for SCOMP Kernel Part I, Release 2.1," 1 October 1984.

5) "Detail Specification for SCOMP Kernel Part II, Release 2.1," 22 October 1984.

6) "TLS to Code Mapping for the SCOMP Kernel Software, Release 2.1," 1 November 1984.

7) "Justification for Unspecified Code for the SCOMP Kernel Software, Release 2.1," 1 November 1984.

8) "Formal Specification for SCOMP Trusted Software, Release 2.1," 24 October 1984.

9) "Detail Specification for SCOMP Trusted Software Part I, Release 2.1," 24 October 1984.

10) "Detail Specification for SCOMP Trusted Software Part II, Release 2.1," 24 October 1984.

11) "TLS to Code Mapping for SCOMP Trusted Software Part II, Release 2.1," 1 November 1984.

12) "Justification of Unspecified Code for SCOMP Trusted Software, Release 2.1," 1 November 1984.

13) "Detail Specification for SKIP Part I, Release 2.0," 29 June 1984.

14) "Detail Specification for SKIP Part II," 10 November 1981.

Hardware Documentation

1) "Detail Specification for SPM Part I, Rev. A," 15 February 1978.

2) "Detail Specification for SPM Part II, Rev. A," 7 March 1979.

3) "SCOMP Study Technical Note: Page Fault Recovery," 15 February 1978.

4) "SCOMP Hardware Verification Report," 28 September 1979.

5) "SCOMP Hardware Verification Plan," 15 April 1978.

6) "SCOMP Page Fault Test Requirements," 17 March 1980.

7) "SCOMP Study Technical Note: SCOMP Unique Instructions," 7 August 1978.

8) "SCOMP Study Technical Note: Argument Addressing Mode Usage on SCOMP," 7 August 1978.

9) "Honeywell Level 6 Minicomputer Systems Handbook," CC71, Rev 0, October 1978.

10) "Honeywell Level 6 Communications Handbook," AT97-02D, May 1981.

11) "Honeywell Custom & Special Products Level 6 SCOMP Maintenance Manual," Document No. 71220265-100, March 1984.

12) "I + II 1822 Asynchronous Communications Line Adaptor (ACLA)," October 1980.

Testing and Configuration Management

1) "Configuration Management Plan for the SCOMP," 20 July 1984.

2) "Trusted Software Test Plan for the SCOMP," 25 July 1984.

3) "Trusted Software Test Report for the SCOMP, STOP Release 2.0," 1984.

4) "Trusted Software Test Report for the SCOMP, Appendix A: Test Programs, Appendix B: Test Results," 1984. 5) "SCOMP Test and Verification Software Description, Rev. 3," 15 April 1980.

6) "SCOMP Kernel Test Procedures," [listings], 1982.

7) "SCOMP Kernel Functional Test Summary," 1982.

8) "Kernel Software Test Report for the SCOMP, Release 2.1, Draft," 1 November 1984.

9) "Test and Verification," April 1982.

10) "SCOMP Page Fault Tests, Rev. 1," March 1980.

Verification

1) "Proving an Operating System Kernel Secure," April 1981.

2) Bonneau, C.H., "Covert Channels in the SCOMP Kernel," revised 10 February 1983.

3) Bonneau, C.H., "Analysis of Failed MLS Proofs for Create_Process and Invoke_Process SCOMP Kernel Functions," 14 February 1983.

4) "SCOMP Verification and the MLS Information Flow Tool, 27 March 1984.

5) "Flow and Covert Channel Analysis for SCOMP Trusted Software, Release 2.1," 19 November 1984.

6) "SCOMP Security Kernel Verification Report, Release 2.0," 21 November 1984.

Miscellaneous

1) "DPS 6 & Level 6 GCOS 6 Assembly Language Reference," CZ38-00A, March 1983.

2) "Honeywell Course H005 Level 6/DPS 6 Assembly Language Student Handbook," July 1982.

3) Bonneau," C.H., "SCOMP C-Compiler Conventions," Honeywell SCOMP- 210, 21 December 1977.

4) "MITRE SCOMP Configuration, Version 2.1," 29 May 1984.

5) Bonneau, C.H., Carnall, J.J., Hall, "SFEP Subsystem Specification," ESD-TR-77-23, October 1976.

6) Boebert, W.E., Bonneau, C.H., Carnall, "Secure Computing, Trends and Applications, Computer Security and Integrity," 1977.

7) Fraim, L.J., "SCOMP: A Solution to the MLS Problem," March 1982.

8) "Multi-level Security on Honeywell SCOMP," n. d. 9) Kert, "Advances in Minicomputer Front-End Security," Presented at Computer Security and Symposium Proceedings, April 1977.

10) Kert, "Role of the Security Kernel in Resource Sharing Systems," n. d.

11) Gilson, J., Mekota, J., "Analysis of Secure Communications Processor Architecture," ESD-TR-351, Vol. 1., November 1975.

12) Bonneau, C.H., "Security Kernel Specification for a Secure Communications Processor," September 1976.

APPENDIX B

TCB Software Functions

Kernel Gates

create_device
create_process
create_segment
delete_segment
*get_device_access
*get_device_status
*get_process_access
*get_process_status
*get_segment_access
*get_segment_status
*get_system_parameters
interrupt_return
invoke_process
lock_secure_terminal
*map_device
map_segment
*mount
*read_system_clock
receive_message
release_process
remove_device
send_message
set_device_access
set_device_status
set_process_access
set_process_status
set_process_subtypes
set_segment_access
*set_segment_status
set_system_clock
shutdown
*sync_segment
unlock_secure_terminal
*unmap_device
unmap_segment
*unmount
unwire_segment
wire_segment

*Callable from ring 3, all other gates callable from within ring 2 only.

Trusted Software

Trusted User Services

Access Control
File Access Modifier
File Display
File Print
Secure Initiator
Secure Server

Trusted Operations Services

Audit Collection
Cancel Terminal Lockout
Change Audit Files
Kernel Bootloader
Message Daemon
Printer Daemon
Secure Loader
Secure Startup
Set Device Access
Set Device Class
Set Time
System Shutdown

Trusted Maintenance Services

Filesystem Check and Repair
Filesystem Restore
Filesystem Save
Make Filesystem
Migrate
Mount Filesystem
Trusted Database Editors
Unmount Filesystem

Trusted SKIP Services

Set Segment Ownership
Trusted Delete

MITRE SCOMP Configuration

Level 6/43 w/SCOMP SPM and VMIU

512K Words Memory

2 Model NDL002 Type BK5B4A Disk Drives
and Controllers

2 8"-Floppy Disk Drives

16 Async Comm Lines

1 Console Comm Line

1 800/1600 BPI Tape Drive

Honeywell SCOMP Configuration

Level 6/53 with control panel

512Kw EDAC memory

2-67Mb mass stores

1-800/1600 bpi tape drive

1-67Mb cartridge module disk drive

2-8 inch floppy disk drives

1 Hard-copy console

1 600 lpm printer

8 async comm lines (up to 9.6Kbps)

1 sync comm line (up to 19.2 Kbpx)

1 L6/L66 Interface

1 1822 ICLA

Appendix C

ANALYSIS OF THE SCOMP REFERENCE MONITOR

The reference monitor of a system is an abstract entity which enforces the authorized access relationships between subjects and objects of that system [43]. As such, the reference monitor is an essential part of computer security (protection) mechanisms. In any system, the access relationships between subjects and objects materialize into references to objects made by the various processes executing instructions on behalf of the human user. The references to objects, which are issued by process instructions, include modifications and retrievals of the object state.

The role of the reference monitor is played, in most systems, by the reference validation (or authorization) mechanisms. These are the principle requirements of the Reference Validation Mechanism (RVM):

- it must be tamper-proof,
- it must always be invoked, and
- it must be small enough to be subject to analysis and tests, the completeness of which can be assured.

The RVM of the SCOMP system is represented within the SCOMP hardware, kernel, and trusted software. Therefore, to demonstrate the SCOMP compliance with the reference monitor concepts, evidence must be provided to demonstrate that both the relevant hardware and kernel mechanisms satisfy the three requirements above.

This section presents the evidence necessary to show that the SCOMP design satisfies the RVM requirements. First, it is noted that SCOMP kernel satisfaction of the RVM requirements depends on the SCOMP hardware. Second, it is argued that the SCOMP hardware design incorporates the necessary mechanisms to support the RVM requirements.

1. Verification of the SCOMP Kernel

The verification of the SCOMP kernel is evaluated elsewhere in this report. Note, however, that the verification of some kernel functions depends on the verification of hardware functions, particularly in the object access area. The analysis of the hardware verification is presented in Appendix D. It is concluded there that the hardware verification is extensive and

thorough, and that despite the use of an informal verification method, sufficient evidence is provided to conclude that the hardware protection mechanisms are sound.

2. The SCOMP Kernel is Always Invoked

In the SCOMP, the user has three basic kinds of objects: user processes, files, and I/O devices. These objects are represented within the SCOMP in terms of kernel objects, such as processes, segments, kernel-private, and other I/O devices. For example, user files are presented as sets of virtual memory segments, which in turn require primary memory segments, disk I/O operations and disk space. The creation/destruction of user objects requires mediation by the kernel, and therefore the kernel is invoked.

Similarly, references to user objects, such as read, write, and execute file are implemented in terms of a series of references to the underlying kernel objects. The kernel-object references are implemented through kernel invocations (e.g., calls to gates, traps, interrupts), or through the execution of hardware instructions. Thus, each reference invokes either the RVM of the kernel or that of the hardware/firmware. The invocation of hardware functions does not imply that the kernel invocation is circumvented. Rather, it means that some of the operations on kernel-provided objects are implemented directly in hardware for efficiency reasons. Nevertheless, these operations should be viewed as kernel operations. Thus, we may conclude that all user references to objects invoke the SCOMP kernel.

3. The SCOMP Kernel is Tamper-Proof (Isolated)

In contrast with the previous two properties of the SCOMP RVM, the "tamper-proof," or "isolation," property is significantly more difficult to demonstrate. The primary reason for this is that most of the mechanisms that ensure the SCOMP kernel is isolated are provided by the hardware. Few methods exist to demonstrate that a hardware design is suitable to support kernel isolation [44]. The reason for this is that hardware considerations have been regarded traditionally outside the scope of verification.

a. Kernel Isolation Properties

In the SCOMP system, the kernel belongs to ring 0 programs and data. Thus, the kernel-area definition, which is necessary to demonstrate that a kernel is isolated [44], is the definition of the ring 0 programs and data structures. Furthermore, the principle isolation property for ring 0 is that of restricting user (instruction) access to memory descriptors, ring registers, and status register of the CPU, SPM, and memory. The isolation

property of the kernel depends on the following hardware mechanisms:

- memory addressing (operand modification) mechanism
- authorization of memory modification
- the ring mechanism: kernel/user separation
- separation between ring 0 and the other rings
- identification of ring 0 - specific instructions
- control of ring 0 entry through calls, traps, and interrupts
- control of ring 0 exit
- authentication of parameters passed to ring 0 by addressing
- modification of access privileges within the access descriptors

(1) Addressable Memory - In SCOMP, memory can be addressed only in four ways: by an instruction execution, by an I/O transfer, by an interrupt or a trap, and by manual operation from the Front Panel. Except in the case of manual operation, all other ways to modify memory require the use of a memory descriptor.

Thus, instructions, I/O transfers, and interrupts/traps cannot reference either memory or devices directly, and therefore cannot gain access to unauthorized kernel programs and data or devices. (Note here that although interrupts and traps run within the kernel, they still use the descriptor mechanism for memory addressing. This helps improve system robustness.) The memory descriptors can be modified only within the kernel, and therefore, dynamic address relocation cannot violate the kernel isolation.

A SCOMP instruction can only modify operands in the current address space; modify processor registers which are not used by the kernel; and transfer from user mode to kernel mode and vice versa (see below) either directly or through traps.

I/O transfers in SCOMP are mediated completely by the descriptor-based mechanism. The user buffer space is separated from the kernel buffer space and user devices are separated from kernel devices. A user device cannot access the kernel except in a controlled way, i.e., through the use of interrupts.

Interrupts and traps make reference to kernel memory and modify device and CPU registers in a controlled way [45, 42]. Since they are part of the kernel they do not violate kernel isolation.

Finally, the Front Panel is controlled by physical security measures. Access to it is strictly limited.

(2) Authorization of Memory Modification - Each memory or device reference issued by an instruction is checked against a descriptor segment and against the rules of the ring mechanism. Each instruction code (a) requires a certain privilege, or certain groups of privileges, to be present within the descriptors, and (b) requires that the ring number associated with the program be within certain kernel-set ring limits [45, 42, 26]. Thus, memory access is controlled by the access privileges within a descriptor and by the ring bracket (limits) mechanism. The ring mechanism may also prevent a user program from invoking the kernel. Consequently, the kernel isolation can be enforced further (this represents an instance of enforcement of the need-to-know principle, i.e., only user programs with a need to invoke the kernel may do so). During indirect addressing, the effective ring number will not be ring 0 unless all indirections are through ring 0 descriptors.

(3) The Kernel/User Separation by the Ring Mechanism -The kernel/user separation within the SCOMP system is accomplished by the separation between ring 0 and the rest of the rings in all processes. This separation is enforced by the following mechanisms:

- Identification of privileged (ring 0) instructions [27].

These instructions may access kernel data structures (i.e., processors, SPM, kernel-device and kernel-memory registers) and perform modifications on them. However, the use of these instructions is restricted to ring 0 code.

- Entry/Exit to/from ring 0: Calls, Interrupts, Traps [46, 45, 42].

The execution of ring 0 instructions is possible only through calls to specific entry points called the kernel gates. Upon such entries some local storage is allocated which remains private to ring 0. That is, no descriptor, or register pointing to that storage, is made available to the outer rings 2 and 3. Furthermore, the ring 0 entry points may not be modified, substituted, or circumvented by the execution of any instruction or by any interrupt or trap. The return address to the calling ring cannot be modified by the caller, and cannot cause return to ring 0 itself for any return operation. Upon return to the

caller ring, the current ring number R=0 is changed to reflect the ring of the caller.

- Parameter Authentication [46].

All parameters passed by reference (address) to ring 0 during calls are authenticated in an uninterruptable way, and are copied in ring 0 memory immediately after authentication. Thus, outer rings cannot supply the addresses which correspond to ring 0 descriptors, or which imply ring 0 privileges. (4) Modifications of Descriptor Privileges - The modifications of the descriptor privileges may only be performed in ring 0. That is, the instructions which invalidate descriptor privileges either belong to ring 0 programs or are privileged instructions. As mentioned above, the rest of the descriptor fields may only be modified by ring 0 programs [26].

4. Conclusions

The SCOMP design and verification documents do include the evidence and informal arguments needed to demonstrate that the SCOMP design implements the reference monitor concept. The proof method outlined in [44] refers to the areas incorporated in the documents and above analysis, and therefore the evidence of SCOMP compliance with the reference monitor concept appears to be sufficient.

Appendix D

SCOMP HARDWARE TESTING AND VERIFICATION

1. The SCOMP Verification Approach

The approach to the verification of the SCOMP hardware is described in references [46] and [47]. The SCOMP designers point out that formal verification has not been attempted. As a consequence, complete assurance cannot be given that all security-relevant aspects of the SCOMP hardware are verified. However, the hardware verification is extensive and thorough. As shown in [26] only a few verification aspects were omitted. The requirement of the A1 evaluation class that an FTLS be provided for the hardware/firmware functions that are visible at the TCB interface is only partially satisfied. Although, the formal specification of hardware/firmware functions in this area are incomplete, the analysis of additional evidence presented by Honeywell showed that the FTLS was consistent with the model. The remainder of this appendix shows how formal and informal techniques were used in this evaluation.

Although the hardware-design verification is the goal of the SCOMP effort, some implementation testing is accomplished in the process. The reason for this is that, while the protection constraints (see section 4 of the report) are derived from design specifications, the verification itself is carried out through the testing of the actual implementation [28]. Thus, some operational assurance is also gained. Design and implementation analysis is carried out whenever testing is impractical or incomplete.

The verification process for the SCOMP hardware consists of three steps. In the first step, all the protection-relevant registers of the architecture, all the operational modes, and all functions (and instructions) which affect each register are listed. The relevant registers and functions (and instructions) are determined from the architectural modules of the security perimeter. Also, all transformations performed by each operational mode and function (or instruction) on each relevant register are defined. Thus, the identification of the security perimeter is the key preliminary step in the SCOMP verification approach.

In the second step, the protection constraints for each register are generated. Note that the relationship between the protection constraints and the protection model is not necessarily defined or stated in this approach. The definition of such a relationship does not appear to be required.

In the third step, all transformations performed on each register by each function are examined in order to establish the consistency between those transformations and those allowed by the protection constraints.

The adequacy of this approach for correctness and completeness verification rests solely on the ability to determine the security perimeter, and to generate complete lists of protection-relevant modules, registers, and protection constraints. The approach does not aid the designer in those two areas because it does not define the above-mentioned relationship with the protection model. In spite of this deficiency, this approach appears to be adequate whenever the protection models are simple and whenever their representation within particular architectures is obvious.

2. Security Perimeter

The preliminary step for verification is definition of the security perimeter of the SCOMP architecture. The security perimeter is defined by the SCOMP hardware configuration and is limited to the Level 6/43 processor with the Security Protection Module with all commercial options known at the time (i.e., 1978-1979), and with all controllers. The configuration excludes the Memory Management Unit which, in the SCOMP, was replaced by the Virtual Memory Interface Unit of the SPM. More specifically, the SCOMP security perimeter is divided into the following three groups of modules: those which are verified through testing and analysis, those that are analyzed but not tested, and those that are neither analyzed nor tested.

3. Module Description and Configuration

Each module within the security perimeter must be specified in a way that makes verification possible. For example, the module description must include the specification of every register and register field, and of every function which either reads or transforms those registers or fields. Furthermore, some assurance must exist that the module description is the one that is actually used in the implementation: i.e., that the module design is frozen.

The importance of the module description and configuration, and of the nature of register and function specifications, cannot be underestimated. The module description and configuration form the basis for the verification process because both the transformations caused by various functions on registers, and the verification (protection) constraints against which the transformations are verified are generated from them. In the SCOMP, the module description provides only a brief functional and physical description of the module verified. The

configuration provides the exact description of the module verified. The configuration description includes module part numbers, design document numbers, drawing numbers, and revision letters.

The register and function specification needed for the verification must be significantly more detailed than, for example, the SPM Function Specification or the SPM Detail Specification [1, 44]. The level of specification detail necessary for the generation of the verification (protection) constraints is similar to that of processor firmware specifications, logic and block diagrams, and/or data flow diagrams. Such specifications appear to have been used in the SCOMP verification process. However, none of these specifications is formal.

4. Verification and Testing Data

The essential part of the verification approach for the SCOMP hardware is included in the verification and testing data [47]. The verification and testing data consist of the following three major parts:

a. Protection Constraints

The "Verify" part (called the protection constraints above) lists the conditions imposed on the registers and on the module functions (or individual instructions) by the correctness and completeness criteria (e.g., by the requirement of consistency with the protection model). The "verify" part is expressed in terms of: module registers and register fields; instructions; functions common to many instructions; predicates; and existential and universal quantifiers.

b. Verification

This part verifies that the transformations performed on the module registers (fields) by various functions (instructions) are consistent with the "verify" part. The verification is performed by testing actual implementation and, in cases when testing is impractical or incomplete, by analysis of the design specifications. Thus, the analysis is also supposed to complement testing in cases of inadequate coverage. The verification part contains the references to the specific tests that are defined in reference [28] and which are performed on the actual hardware.

c. Conclusions

This part contains the description of the verification results, the assumptions made during the tests, and the

cautionary notes that must be observed by the operating system and kernel designers. For example, the ENT instruction which may return from ring Rcur=0 to ring R3 without changing the stack frame (i.e., the T register) must not be used by any operating system software.

5. Summary

The informal verification approach used for the SCOMP hardware is extensive and thorough. In the area of protection-mechanism verification, it provides sufficient evidence that the SCOMP hardware design forms a sound basis for the development of a security kernel. Thus, it can be shown that the SCOMP hardware design satisfies the requirements of the reference monitor concept [4].

However, it must be noted that the state of the art in the verification of the hardware-supported protection mechanisms is such that it is difficult to guarantee that no design flaws are left undiscovered. The SCOMP verification approach is no exception. It presents a number of problems characteristic to the state of the art such as: incomplete formal top-level specification of the hardware/firmware functions that are visible at the TCB perimeter; lack of clear distinction between design and implementation verification; lack of an explicit relationship between the verification data and the protection model (which usually results from lack of precise definition of the security perimeter); and lack of assessment of the analysis and test coverage. In fact, in reference [26] it is shown that neither the design nor the implementation testing and analysis cover completely all possible cases.

Lack of complete coverage in design verification (and in implementation testing) does not necessarily imply that the SCOMP design/implementation is flawed. However, lack of complete coverage requires that confidence in the hardware design/implementation be gained in alternate ways: e.g., by discussions with the systems designers, by careful review of all possible implications of the verification omissions, and by penetration analysis. All concerns raised with the system designers were answered satisfactorily.

REFERENCES

1. Vickers-Benzel, T. "Overview of the SCOMP Architecture and Security Mechanism," MITRE Technical Report, MTR 9071, September 1983.

2. Fraim, L. J.,"Multi-Level Security on Honeywell SCOMP," Honeywell, Inc., 1982.

3. Graham, R. M., "Protection in an Information Processing Utility," CACM, Vol. 11, No. 5, May 1968.

4. Bonneau, C. H., "Secure Communications Processor (SCOMP) Study Technical Note, SCOMP-Unique Instructions," Honeywell, Inc. Avionics Division, 1978.

5. U.S. Department of Defense "Trusted Computer System Evaluation Criteria," CSC-STD-001-83, 15 August 1983.

6. SCOMP Trusted Facility Manual, STOP Release 2.1, Honeywell Information Systems, Inc., 1 November 1984.

7. SCOMP User's Reference Manual, STOP Release 2.1, Honeywell Information Systems, Inc., 1 November 1984.

8. Bonneau, Charles H., "Covert Channels in the SCOMP Kernel," revised 10 February 1983.

9. Detail Specification for SCOMP Kernel Part II, Release 2.1, Honeywell Information Systems, Inc., 22 Ocotober 1984.

10. Detail Specification for SCOMP Trusted Software Part II, Release 2.1, Honeywell Information Systems, Inc., 24 October 1984.

11. SCOMP Trusted Computing Base, Honeywell Information Systems, Inc., 25 July 1984.

12. Schaefer, M., et al., "Program Confinement in KVM/370," Proc. of ACM National Conference, Seattle, October 1977.

13. Denning, D.E., and P.J.Denning, "Certification of Programs for Secure Information Flow," CSD-TR 181, Purdue Univ., March 1967.

14. SCOMP Kernel Test Procedures, [listings], Honeywell Information Systems, Inc., 1982.

15. SCOMP Kernel Functional Test Summary, Honeywell Information Systems, Inc., 1982.

16. Kernel Software Test Report for the SCOMP, Release 2.1, Honeywell Information Systems, Inc., Draft, 1 November 1984.

17. Trusted Software Test Plan for the SCOMP, Honeywell Information Systems, Inc., 25 July 1984.

18. Trusted Software Test Report for the SCOMP, STOP Release 2.0, Honeywell Information Systems, Inc., 1984.

19. Trusted Software Test Report for the SCOMP, Appendix A: Test Programs, Appendix B: Test Results, Honeywell Information Systems, Inc., 1984.

20. Linde, R. R., "Operating System Penetration," Proceedings of the National Computer Conference 1975, AFIPS Press, Montvale N.J., Vol. 44, pp. 361-368.

21. DoD Directive 5200.28, "Security Requirements for Automatic Data Processing (ADP) Systems," revised April 1978.

22. Bell, D.E., and L.J. LaPadula, "Secure Computer Systems," ESD-TR-73-278, Vol. I-III, The MITRE Corp., Bedford, MA, November 1973 - June 1974.

23. Detail Specification for SCOMP Kernel Part I, Release 2.1, Honeywell Information Systems, Inc., 1 October 1984.

24. Detail Specification for SCOMP Trusted Software Part I, Release 2.1, Honeywell Information Systems, Inc., 24 October 1984.

25. Detail Specification for SPM Part I, Rev. A, Honeywell Information Systems, Inc., 15 February 1978.

26. Detail Specification for SPM Part II, Rev. A, Honeywell Information Systems, Inc., 7 March 1979.

27. Top Level Specification for SCOMP Kernel Software, Release 2.0, Honeywell Information Systems, Inc., 29 June 1984.

28. Formal Specifications for SCOMP Trusted Software, Release 2.1, Honeywell Information Systems, Inc., 10/24/84.

29. SCOMP Interpretation of the Bell-LaPadula Model, Honeywell Information Systems, Inc., 25 October 1984.

30. TLS to Code Mapping for the SCOMP Kernel Software, Release 2.1, Honeywell Information Systems, Inc., 1 November 1984.

31. TLS to Code Mapping for SCOMP Trusted Software, Release 2.1, Honeywell Information Systems, Inc., 1 November 1984.

32. SCOMP Security Kernel Verification Report, Release 2.0, 21 November 1984.

33. Feiertag, Richard J., "A Technique for Proving Specifications are Multilevel Secure," Computer Science Lab. Rep. CSL-109, SRI International, 10 January 1980.

34. Justification for Unspecified Code for the SCOMP Kernel Software, Release 2.1, Honeywell Information Systems, Inc., 1 November 1984.

35. Justification of Unspecified Code for SCOMP Trusted Software, Release 2.1, Honeywell Information Systems, Inc., 1 November 1984.

36. Configuration Management Plan for the Secure Communications Processor SCOMP, Honeywell Information Systems, Inc., 20 July 1984.

37. Flow and Covert Channel Analysis for SCOMP Trusted Software, Release 2.1, Honeywell Information Systems, Inc., 19 November 1984.

38. Fraim, Lester J., "SCOMP: A Solution to the Multilevel Security Problem," Computer, Vol.16, No.7 (July 1983), pp.26-34.

39. Bell, D.E., and L.J. LaPadula, "Computer Security Model: Uni-fied Exposition and MULTICS Interpretation," ESD-TR-75-306, The MITRE Corp., Bedford, MA, June 1975, (AD A023588).

40. SCOMP Test and Verification Software Description, Rev. 3, Honeywell Information Systems, Inc., 15 April 1980.

41. Honeywell Level 6 Minicomputer Systems Handbook, CC71, Rev 0, October 1978.

42. Honeywell Level 6 Communications Handbook, AT97-02D, May 1981.

43. Honeywell, Inc., "Secure Communication Processor-Hardware Verification Plan," Draft Report, Program Code No. 7P10, prepared for Contract No. NAVELEX N00039-77-C-0245.

44. Carnall, J.J. and Wright, A.F., "Secure Communication Process-Hardware Verification Report," Technical Report, Honeywell Inc., Program Code No. 7P10, prepared for Contract No. NAVELEX N00039-77-C-0245.

45. Gligor, V.D., "Analysis of the Hardware Verification of the Honeywell SCOMP," Technical Report prepared for the DoD Computer Security Center under contract No. MDA-904-81-G-0012 at the University of Maryland, College Park, April 1983.

46. Anderson, J.P., "Computer Security Technology Planning Study," ESD-TR-73-51, Vol. I, ESD/AFSC, Hanscom AFB, Bedford, Mass., October 1972 (NTIS-AD-758 206).

47. Millen, J., "Kernel Isolation for the PDP-11/70," Proc. of the IEEE Symposium on Security and Privacy," Oakland, California, 1982.

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402

END

DTIC

5-86